Proceedings of the First ERCIM Workshop on eMobility

WULFF, Markus

KONSTANTAS, Dimitri (Ed.), BRAUN, Torsten (Ed.), MASCOLO, Saverio (Ed.)

Abstract

This volume contains all accepted papers of the ERCIM Workshop on eMobility, which has been held in Coimbra, Portugal, on May 21, 2007. Papers from three main areas have been selected for the workshop. The workshop papers discuss several topics of the ERCIM eMobility working group, namely – Traffic engineering and mobility management – Wireless (sensor) networks – Pervasive computing and mobile applications. The goal of the ERCIM workshop is to foster collaborative work within the European research community and to increase co-operation with European industry. Current progress and future developments in the area of eMobility should be discussed and the gap between theory and application should be closed.

Preface

This volume contains all accepted papers of the ERCIM Workshop on eMobility, which has been held in Coimbra, Portugal, on May 21, 2007. Papers from three main areas have been selected for the workshop. The workshop papers discuss several topics of the ERCIM eMobility working group, namely

- Traffic engineering and mobility management
- Wireless (sensor) networks
- Pervasive computing and mobile applications.

The goal of the ERCIM workshop is to foster collaborative work within the European research community and to increase co-operation with European industry. Current progress and future developments in the area of eMobility should be discussed and the gap between theory and application should be closed.

At this point, we want to thank all authors of the submitted papers and the members of the international program committee for their contribution to the success of the event and the high quality program. In a peer review process, 12 papers have been selected out of 21 submissions. The reviewers evaluated all the papers and sent the authors the comments on their work.

The workshop started with a keynote from Prof. Luis M. Correia on "eMobility: present and future challenges", which formed a great introduction to the following presentations.

Coimbra, one of the oldest and most famous universities in Portugal and Europe gave us an inspiring environment for a lot of intensive and fruitful discussions. The next ERCIM workshop is scheduled for 2008. We hope that a lot of our participants and many new colleagues will take this opportunity to continue exchanging their knowledge and experiences devoted to the development and use of eMobility.

Torsten Braun
Dimitri Konstantas
Saverio Mascolo
Markus Wulff

General chairs

Torsten Braun, University of Bern, Switzerland
Dimitri Konstantas, University of Geneva, Switzerland

TPC chairs

Saverio Mascolo, Politecnico di Bari, Italy
Markus Wulff, University of Bern, Switzerland

Technical program committee

Francisco Barceló-Arroyo, Universitat Politecnica de Catalunya, Spain
Hans van den Berg, University of Twente, The Netherlands
Raffaele Bruno, Italian National Research Council, Italy
Giovanni Giambene, University of Siena, Italy
Geert Heijenk, University of Twente, The Netherlands
Jean-Marie Jaquet, University of Namur, Belgium
Andreas J. Kassler, Karlstad University, Sweden
Yevgeni Koucheryavy, Tampere University of Technology, Finland
Edmundo Monteiro, University of Coimbra, Portugal
Maria Papadopouli, University of Crete, Greece
Antonio M. Peinado, University of Granada, Spain
Vasilios Siris, University of Crete, Greece
Dirk Stähle, University of Wuerzburg, Germany
Do van Thanh, NTNU, Trondheim, Norway
Mari Carmen Aguayo Torres, University of Malaga, Spain
Vassilis Tsoussidis, Democritus University of Thrace, Greece
Jean-Frédéric Wagen, College of Engineering and Architecture of Fribourg, Switzerland

Table of Contents

Impact of Feedback Channel Delay on Adaptive OFDMA Systems

D. Morales-Jiménez, Juan J. Sánchez, G. Gómez. M. Carmen Aguayo-Torres, J. T. Entrambasaguas *

Departamento de Ingeniería de Comunicaciones, Universidad de Málaga, Spain,
<morales,jjsanch,ggomez,aguayo,jtem>@ic.uma.es

Abstract. Most standards for the forthcoming beyond 3G (B3G) and 4G technologies state Orthogonal Frequency-Division Multiple Access (OFDMA) as a very promising candidate to be used as a digital modulation scheme. OFDMA combines multiple access techniques with Adaptive Quadrature Amplitude Modulation (AQAM) to maximize system performance while keeping the errors below a certain target. In order to achieve this objective, Channel Quality Indicators (CQI) are fedback from the receivers. However, potential delays in the reception of such CQIs may lead to a system performance degradation. This paper analyzes the impact of CQI feedback delay over a Long Term Evolution (LTE) network.

1 Introduction

Currently, the Third Generation Partnership Project (3GPP) is working on the evolution of the 3G Cellular Networks standardization process [1]. A collaborative process that involves operators, manufacturers and research institutes is in progress to discuss views and proposals on the evolution of the Universal Terrestrial Radio Access Network (UTRAN). LTE specifications are targeting to become a high-data-rate, low-latency and packet-optimized radio-access technology [2]. LTE multiple access in the downlink is based on Orthogonal Frequency-Division Multiple Access (OFDMA), which is a promising technique to provide an efficient access over high-speed wireless networks [3]. LTE will offer broadband wireless access at data rates of multiple Mbit/s to the end-user and within a range of several kilometers. OFDMA at the physical layer, in combination with a Medium Access Control (MAC) layer, provides an optimized resource allocation and Quality of Service (QoS) support for different types of services.

High spectral efficiency in OFDMA environments is achieved by dividing the total available bandwidth into narrow sub-bands to be shared by users in an efficient way. Besides, Adaptive Quadrature Amplitude Modulation (AQAM) is also used to maximize the transmission efficiency while keeping the Bit Error

* This work is partially supported by the Spanish Government and the European Union under project TIC2003-07819 (FEDER) and by the company AT4Wireless, S.A.

Rate (BER) below a desired target. These techniques require the transmitter to be instantaneously channel-aware so that proper modulation schemes and frequency sub-bands are selected dynamically. Thus, the transmitted signal is continuously adapted to the varying channel conditions.

In order to select the modulation scheme for each subcarrier, the channel has to be known at the transmitter. With this objective, Channel Quality Indicators (CQI) are fedback from the receivers to the transmitter. However, potential delays in the reception of CQI through the feedback channel may cause a system performance degradation. Impairments in adaptation due to the delayed reception of CQI were analyzed in [4] for a generic AQAM system. Such delay is a further undesirable effect as mobile terminal speed increases (since channel time-coherence is shorter). Currently, the possibility to concatenate multiple sub-frames into longer Transmission Time Interval (TTI) is being considered in LTE in order to reduce the signalling overhead. However, this would mean a longer delay in the modulation adaptation process.

In this work, a model based on LTE specifications [2] has been implemented on top of WM-SIM [5] in order to evaluate the maximum admissible delay of feedback channel. The model allows to simulate the LTE downlink where CQI is fedback from each User Equipment (UE) to the Enhanced Node-B (eNodeB).

The rest of the paper is structured as follows. In section 2 a brief description of the LTE is presented, focusing on both OFDMA and AQAM techniques and how CQI delays affect performance in this kind of systems. The scenario under study and a description of the implemented system can be found in section 3, whereas simulation results are shown in section 4. Finally, section 5 gathers the main conclusions and future work.

2 OFDMA overview in LTE systems

Orthogonal Frequency Division Multiplexing (OFDM) is a modulation technique widely used to counteract the effects of Inter Symbol Interference (ISI) in frequency selective channels [6]. OFDM divides the transmission band in a large number of sub-bands narrow enough to be considered flat. The symbol sequence is split into lower speed symbol streams transmitted simultaneously on the resulting comb of carriers. Specifically in LTE, several transmission bandwidths from 1.25 to 20 MHz are defined, with a corresponding number of subcarriers in the range from 128 to 2048. At the transmitter, several subcarriers are employed to locate reference symbols in order to allow channel measurements. For the 20 MHz bandwidth mode, the number of useful subcarriers is reduced to 1200 whereas 848 subcarriers are guards and pilots symbols are transmitted over 200 of them.

An Inverse Fast Fourier Transform (IFFT) efficiently performs the modulation process. Its reciprocal process, the forward Fast Fourier Transform (FFT), is used to recover the data as a cyclic extension of the OFDM symbol eliminates the residual ISI. In this way, OFDM can be considered as a time-frequency squared pattern, where each bin can be addressed independently.

Modulation of the OFDM subcarriers is analogous to that of the conventional Single Carrier (SC) systems. Supported downlink data-modulation schemes in LTE are QPSK, 16QAM, and 64QAM. The number of bits allocated to each subcarrier can be modified on a sub-frame basis to simultaneously track the time variant frequency response of the channel and fulfill the BER service requirements. In LTE, the minimum downlink Transmission Time Interval (TTI) corresponds to the sub-frame duration, $T_{sub-frame} = 0.5$ ms, and a sub-frame is composed by a signalling symbol and six data symbols. In addition, a second frame structure is also supported with the intention of providing co-existence with LCR-TDD (Low Chip Rate - Time Division Duplexing). With this alternative frame structure, the sub-frame is enlarged up to 5 ms.

When OFDM is also used as multiplexing technique, the term OFDM Access (OFDMA) is preferred. In this case, a block of bins is assigned to a single user in what can be considered a hybrid TDMA-FDMA technique. In LTE, radio blocks consists of $M = 12$ subcarriers assigned along a sub-frame. With the channel information obtained from the pilot symbols, Channel Quality Indicators (CQI) are estimated at the receiver and fedback to the transmitter for modulation adaptation and resource allocation purposes. Although both block-wise transmission (localized) and transmission on distributed sub-carriers are to be supported in LTE, in this work only blocks consisting of contiguous subcarriers have been considered as fast adaptive modulation is more sensible to the adaptation delay.

2.1 Effects of adaptation delay

In the adaptation process two different impairments can be identified [4]. First, modulation selection is performed using noisy Channel State Information (CSI) as exact channel estimation is not possible. Moreover, Doppler shift may cause different CSI at the time of transmission from that at the time of channel estimation. This work focuses on the second degradation as it is definitively determined by design aspects which can not be modified in prototyping time.

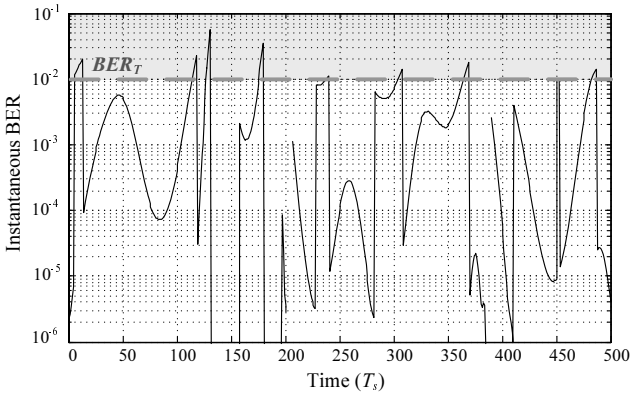


Fig. 1. Instantaneous BER evolution with feedback delay.

The impact of adaptation delay on the instantaneous BER for a single subcarrier is illustrated in Fig. 1. Delay on adaptation may cause a wrong decision in the modulation level at the transmitter and, hence, predefined BER requirements may be unfulfilled. Fig. 1 represents the instantaneous BER as a function of time (normalized to OFDM symbol period T_S). It is shown how the instantaneous BER values are above the target BER ($BER_T = 10^{-2}$) during short time intervals. These intervals corresponds to those when the selected modulation scheme does not match the channel conditions due to the delay on CQI report.

3 System Model

The downlink direction of an OFDMA wireless system has been studied. As shown in Fig. 2, an evolved Node B (eNode B) is connected to one or several User Equipments (UE) through a radio channel. In this example, channel conditions for UE_1 and UE_2 are different since they are located in distinct places and have different speeds. Therefore, each of them report a different CQI values (CQI_1 and CQI_2) to the eNode B. This information about channels conditions will be taken into account to allocate radio resources for each UE.

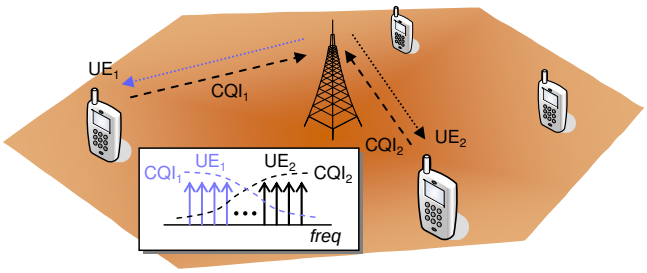


Fig. 2. Scenario under analysis.

The scenario under analysis is modelled using WM-SIM. A block diagram of the implemented OFDMA model is shown in Fig. 3. Model includes the following subsystems: a traffic generator that produces the information flows associated to each user; an eNode B, which implements the main PHY/MAC functionalities at the radio interface; a Rayleigh frequency-selective radio channel; a set of user equipments in charge of processing adequately the received signal; and finally, a Quality of Service (QoS) metrics functionality that collects performance statistics from the simulations (BER, delay, throughput and loss rate in the transmission queues).

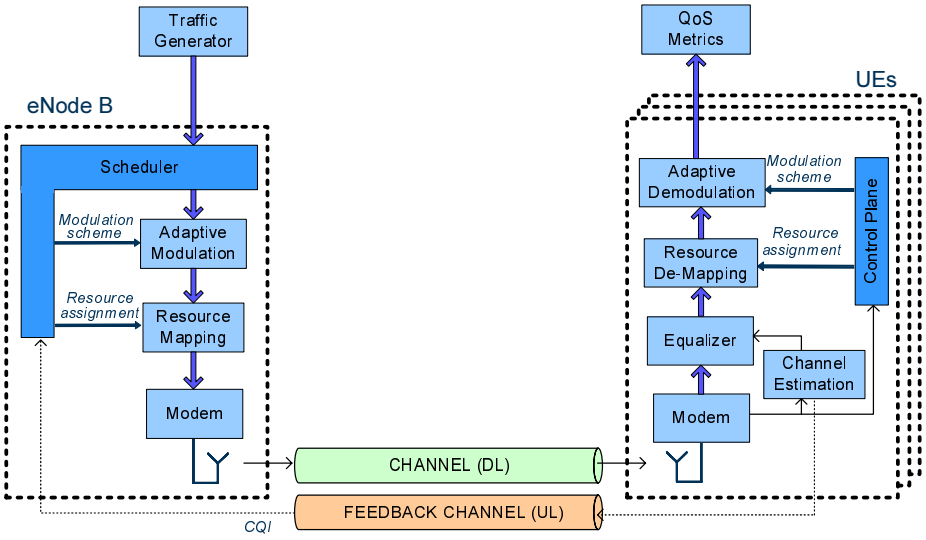


Fig. 3. Downlink OFDMA Wireless System model.

3.1 Enhanced Node B

The eNode B subsystem aims to simulate the basic functionalities of a base station that uses OFDMA technology. This subsystem is made up by four different blocks (as shown in Fig. 3):

– *Scheduler.* Incoming information flows from the traffic generator are stored into N_u First-In First-Out (FIFO) queues (one per user). The cross-layer scheduler is responsible for allocating transmission turns to users following a certain algorithm. Allocation criteria is based on various information like the Channel Quality Indicators (CQI) and/or the queues occupancy. Calculation of CQI is performed for each sub-band (group of 12 consecutive subcarriers) by averaging the SNR value all over the chunk. Once the transmission turn is allocated, a number of bits (according to the UE and chunk modulation level) are extracted from the corresponding queue.

– *Adaptive Modulation.* The modulation level for each user is selected at a subframe rate, according to their estimated instantaneous SNR and target BER values. Instantaneous SNR (γ) is received at the eNode B from each UE through a feedback channel that introduces a configurable transmission delay. No losses are assumed in feedback channel since their effects are beyond the objectives of this study. Adaptive modulation is carried out by means of predefined SNR thresholds that select the proper modulation level $m(\gamma)$ depending on the BER_T, as shown in Fig. 4. Once the scheduler has selected a particular user, the sequence of bits extracted from the queues are

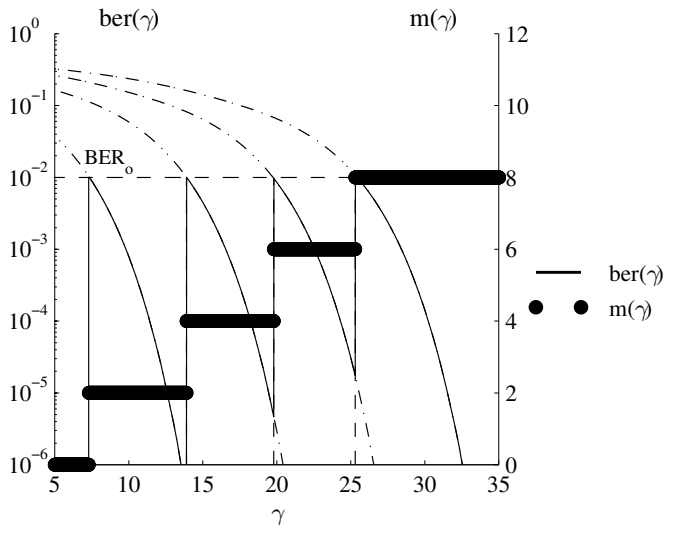


Fig. 4. SNR Thresholds for Adaptive Modulation.

mapped onto their corresponding constellation. Therefore, different constellations can be used along the OFDM symbol since the information conveyed for each user may has a different modulation level.

– *Resource Mapping.* According to the scheduling decision, complex data symbols are mapped on to a certain physical resource block, corresponding to a particular time-frequency area. Thus, the frequency selectivity of the channel can be alleviated. In addition, OFDM symbols are fully conformed, including reference symbols (pilots) and guard periods.

– *Modem.* The transmission modem performs several actions before transmitting the signal to the radio interface. Firstly, the Inverse Fast Fourier Transform (IFFT) is applied in order to convert the OFDM symbol to the time-domain. Secondly, a cyclic prefix is appended to the OFDM symbol to avoid ISI and to minimize temporal synchronization problems between transmitter and receiver.

3.2 Radio Channel

Downlink and uplink radio channels have been modelled in a very different way. While the downlink channel includes a complete frequency-selective multipath model, the uplink scenario (feedback channel) has been simplified in order to focus on the delay effect.

Downlink Radio Channel. A frequency-selective channel is modelled, considering the temporal fading due to multipath propagation [3]. Channel response

is assumed to be composed by multiple taps with predefined delays and mean power. This multi-tap configuration determines the mean power profile, which has been set according to the Suburban Macro scenario defined in [7]. Temporal variations on this profile follow a Rayleigh distribution that affects the instantaneous taps power, while taps delays are assumed to be constant.

Additionally, downlink channel includes the effect of noise, modelled as Additive White Gaussian Noise (AWGN) with zero mean at the receive antenna. Noise variance depends on the pre-configured SNR value since constant transmit power has been assumed.

Feedback Channel. As the main purpose of this paper is to analyze the impact of the feedback channel delay on adaptive OFDMA systems, this feedback channel has been modelled as a perfect delay line. This simplification allows to isolate other undesirable effects from the results. Hence, uplink channel has been just modelled as a FIFO queue, which introduces a configurable delay to the CQIs.

3.3 User Equipment Subsystem

Each UE is modelled as an independent subsystem, which processes its received signal through the following blocks sequence (see Fig. 3):

- *Modem* This block receives the transmitted physical signal after being affected by the radio link between the eNode B and a particular UE. Cyclic extension introduced by the eNode B is removed, and afterwards, Fast Fourier Transform (FFT) is applied to recover the received OFDM symbol into frequency-domain.
- *Channel Estimation.* Each UE estimates its correspondent channel frequency response as well as the instantaneous SNR of its received signal. Ideal channel estimation has been assumed in order to isolate the effects of feedback channel delay.
- *Equalization.* The estimated channel frequency response is used to compensate the undesirable effects of the radio channel on the received OFDM symbol. In this block, zero-forcing is adopted as equalization technique.
- *Control Plane.* Scheduling signalling information is extracted from the first and second symbols of a subframe. This information is needed by the Resource De-mapping and Adaptive Demodulation functionalities.
- *Resource De-mapping.* Received OFDM symbols are de-mapped according to the scheduling signalling information. Once a subframe is completed, the data segments allocated to each UE are identified.
- *Adaptive Demodulation* Data segments from each user are demodulated according to the received control information.

4 Simulation Results

Main simulation parameters are listed in Table 1. Different UE speeds have been simulated in order to identify the maximum speed that fulfill the predefined QoS requirements. The users speed vary from 5 Km/h (pedestrian) to 30 Km/h. Higher UE speeds implies faster temporal changes in channel response and, as a consequence, the influence of the feedback channel delay on the transmission adaptation will be greater. On the contrary, CQIs from UEs at lower speeds (i.e. experiencing slow varying channels) will be even less affected by the feedback delay.

Table 1. Configuration parameters

Parameter	Value
FFT Size	2048
Data Sub-carriers	1200
Cyclic prefix length	144 samples
Carrier Frequency	1.8 GHz
Sampling Frequency	30.72 MHz
UE Speed	5-30 Km/h
Feedback Delay	1-5 ms
Target BER	10^{-2} and 10^{-3}

Figure 5 illustrates the effect of feedback channel delay on the average BER for different UE speeds and same target BER (10^{-2}). For a UE speed of 5 Km/h (a), channel response has a very slow variation and therefore, feedback channel delay does not affect significantly (BER values remain under the target even for

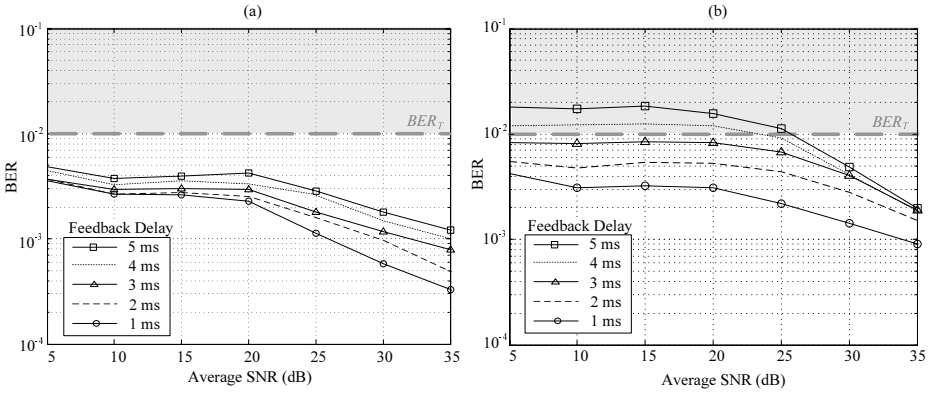


Fig. 5. BER vs. average SNR for different feedback channel delays for: (a) UE speed: 5 Km/h; (b) UE speed: 15 Km/h.

5 ms delay). However, it is clear how results get worse as the delay increases. When UE moves faster (15 Km/h) (b), the impact of delay is greater and there is an important performance degradation. The maximum admissible delay for the feedback link is about 3 ms when the UE moves at 15 Km/h. Shadowed area in the figure represents those BER values above BER_T.

In Figure 6, the average BER is presented as a function of feedback channel delay for different UE speeds and average SNR of 20 dB. In case (a), BER results are always below the $BER_T = 10^{-2}$ for quasi-pedestrian speeds (5 and 10 Km/h). However, for higher UE speeds, BER_T is exceeded even for small delays: 1.5 ms is the maximum admissible delay at 30 km/h.

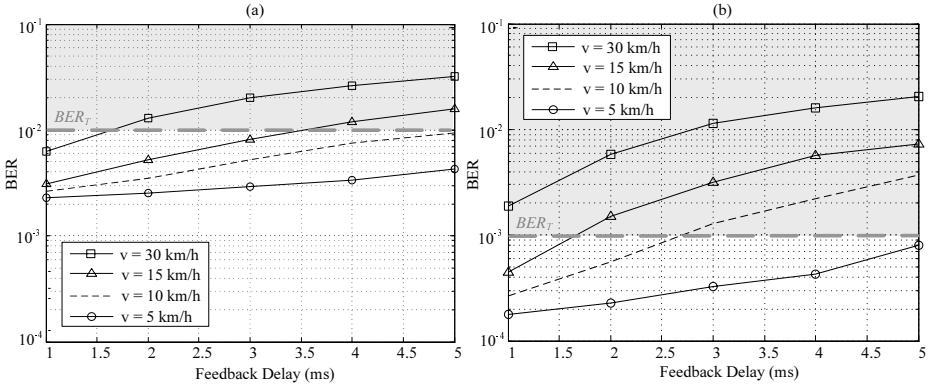


Fig. 6. BER vs. feedback delay when target BER value is 10^{-2} (a) and 10^{-3} (b) for different UE speeds.

For a more restrictive and reliable constraint, e.g. $BER_T = 10^{-3}$, BER requirements are only fulfilled by pedestrian UEs (5 Km/h). When UEs speed is higher (from 10 Km/h on) even a small delay causes a BER higher than the target value (e.g. 2.75 ms at 10 Km/h and 1.75 ms at 15 Km/h).

In the forthcoming B3G or 4G cellular systems, UE speed will be limited in most of the cases to pedestrian values. For this situation, it has been shown how delays in feedback channel are not so restrictive, and system performance does not experience a degradation even for delays up to 5 ms.

5 Conclusions and future work

Along this paper, the impact of CQI feedback delay on an OFDMA system is addressed. As it was foreseen, adaptive modulation is very sensitive to such delays. They may cause a wrong selection of the instantaneous modulation scheme since the CQI used in that selection may not match current channel conditions.

Simulation results show that a system performance degradation is detected for pedestrian speeds (5 Km/h) when feedback channel delays are above 5 ms.

However, BER results are kept under the target value if delays are below 5 ms even for a BER_T of 10^{-3}. When UE speed is higher, channel time coherence is lower, i.e. temporal correlation decreases. Hence, CQI information becomes outdated sooner and average BER results are below the specific target only for low feedback delays.

A complementary study focused on the effects of errors in CQI information for a given delay is an ongoing work. All this work will lead finally to the development of algorithms to estimate and compensate both potential delays and errors in the feedback link. These algorithms will make possible to keep system performance even for non-ideal feedback channel conditions.

References

1. 3rd Generation Partnership Project (3GPP). www.3gpp.org
2. 3rd Generation Partnership Project. "UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE)". http://www.3gpp.org/Highlights/LTE/LTE.htm
3. Goldsmith, A.: Wireless Communications. Cambridge University Press, 2005.
4. Paris, J.F., Aguayo-Torres, M.C., Entrambasaguas, J.T.: Non-ideal Adaptive Modulation: Bounded Signaling Information and Imperfect Adaptation. Proceedings of the Global Communication Conference, Globecom 2004, Dallas, December, 2004
5. Juan J. Sánchez, G. Gómez, D. Morales-Jiménez. J. T. Entrambasaguas: "Performance evaluation of OFDMA wireless systems using WM-SIM platform". MOBIWAC. Proceedings of the international workshop on Mobility management and wireless (Torremolinos, Spain.). ISBN:1-59593-488-X.2006, Pages: 131 - 134.
6. Nee, R., Prasad, R.: OFDM for wireless multimedia communications. Artech House Publishers, Boston, 2000
7. 3rd Generation Partnership Project. "Technical Specification Group Radio Access Network; Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA)", Release 7. Technical Specification 25.814, http://www.3gpp.org.

Impact of the Variance of Call Duration on Teletraffic Performance in Cellular Networks

Antonietta Spedalieri, Israel Martin-Escalona, Francisco Barcelo-Arroyo
Dept. of Telematic Engineering
Technical University of Catalonia
Barcelona, Spain
aspedali@entel.upc.es, imartin@entel.upc.es, barcelo@entel.upc.es

***Abstract*— The aim of the paper is to study the impact of mobility on the relationship between the variance of the call duration and performance metrics. Teletraffic and Quality of Service (QoS) variables of a WCDMA (Wideband CDMA) mobile network are considered as quality measurements. In order to achieve this characterization, a simulation tool has been developed and used. The simulator implements the performance of a UMTS (Universal Mobile Telecommunication System) network. This paper deals, on one hand, with the impact on Teletraffic variables of progressive cellular division and, on the other hand, with the characterization of the impact of variations in voice service second moment parameter, using lognormal distributed service time in spite of exponential one normally used in literature. The consequences of introducing a lognormal distribution for the call duration instead of the classical exponential one are analyzed as well as the impact of enhancing second moment of call holding time lognormal distribution.**

***Index Terms*— QoS, Traffic Modeling, Channel Holding Time, Handoff Area.**

1. Introduction

WCDMA mobile networks like UMTS are designed for flexible delivery of any type of service such as speech, video telephony, images and multimedia, browsing, audio and video streaming etc, where each new service does not require a particular network optimization. In addition to flexibility, the WCDMA radio solution brings advanced capabilities in service requirements differentiation such as adaptable bit rate, a different connection time and network resources occupation, low delays, quality of service (QoS) differentiation and, at the same time, interworking with existing networks (such as GSM/GPRS).

Moreover, progressive cellular division introduces changes in the Teletraffic analysis that oppose the previous studies carried on fixed network and even in the first mobile

cellular networks based on larger cells. The fact that the latest mobile network designs are based on smaller cell size has, as direct consequence, the increase of the number of handoffs produced by a call and the presence of new processes bounded to the new mobility concept.

In a mobile network call blocking probability and handoff blocking probability are important QoS variables and can be used to characterize the quality of voice service and then extended to general quality of other service types. Moreover, variables such as channel holding time, the time between two consecutive handoff processes, the handoff duration, the time within the handoff area, the number of request of a single handoff process, etc, must be taken into account in the Teletraffic analysis. All these variables can be useful for a better network planning.

The channel holding time has been deeply studied using several approaches. Thus, [1 and 2] propose an analytical study of this variable, while [3] provides a characterization of it using a simulation procedure and [4] does the same by means of a field-data analysis. On the other hand, variables related with the handoff process have been rarely considered in the recent Teletraffic research [5-8]. However, since the handoff rate has become higher along with the reduction of the cell size, their importance has increased. The reference studies related with these variables consider only Fixed Channel Allocation (FCA) networks [1-8].

This paper deals, on one hand, with the impact on Teletraffic variables of progressive cellular division and, on the other hand, with the characterization of the impact of variations in voice service second moment parameter, using lognormal distributed service time in spite of exponential one normally used in literature. The consequences of introducing a lognormal distribution for the call duration instead of the classical exponential one are analyzed following findings made in recent studies, which suggested that a lognormal distribution gives a more realistic fit of call duration [9, 10]. Using lognormal distribution, we give the right importance to short calls, that especially affect signalization channels, and to calls that do not last longer than the mean holding time. Enhancing second moment of call holding time lognormal distribution, larger calls presence probability grows and impact on the system can also be evaluated.

In this study, three Teletraffic variables have been considered: channel holding time, time between handoffs and handoff duration. In addition, the study shows results of QoS (i.e. call blocking probability and forced interruption probability).

The paper is organized as follows. The simulation tool is presented in section 2. In section 3 all variables analyzed in the study are described in detail. Sections 4 and 5 analyze the obtained results about the impact of variations in service time variance parameter on QoS and Teletraffic variables respectively. Finally, section 6 summarizes some key achievements and conclusions.

2. Simulator Description

The simulator presented in [7] has been modified and improved. The substance of simulation tool is basically the same as described in [7], in this paragraph we point out only the substantial differences.

In our model, speech traffic at 12.2 kbps is delivered with a spreading factor of 256. First improvement, has been to consider 36 tri-sectorial cells organized in a toroidal domain with calls moving uniformly overall the simulation area. Cell dimension has been calculated using the link budget of the WCDMA uplink (to calculate the maximum allowed propagation loss) and Okumura-Hata propagation model. Statistics are taken on the overall simulation area. The assumptions that have been taken in the link budget for the receivers and transmitters are shown in Table I.

TABLE I. ASSUMPTIONS FOR MOBILE STATION AND BASE STATION

Mobile Station	
Maximum Transmission Power	21 dBm
Antenna Gain	0 dBi
Body Loss	3 dB
E_b/N_0 requirement	6.4 dB
Base Station	
Maximum Transmission Power	43 dBm
Antenna Gain	18 dBi
E_b/N_0 requirement	5.5 dB

The whole-call duration and the fresh call arrival process are the only traffic inputs to the simulator. No other consideration has been taken regarding channel holding time or HO process: those variables are a result of the traffic, type of environment, cell size, mobility pattern, signal power received by each station, considered load etc. This provides the simulator a very realistic scope.

An offered load of 50, 70 and 90% has been considered. Offered traffic for each load, input parameter to the simulator as inter arrival time, is showed in Table II. Each call remains active for a period of time which can be exponentially or log-normally distributed with mean 120 seconds and, according the aim of the study, different values for squared coefficient of variation are considered for a presented results set. Following [10] and [13], squared variation coefficients of 1, 3 and 10 have been considered.

The initial direction of a mobile station (MS) follows a uniform distribution and the speed of the mobile users is Gaussian-distributed. The following movement pattern has been considered: a medium speed suburban pattern with mean 13.9 meters/second and standard deviation of 4 meters/second (considering in-car losses). The MSs change their speed and direction while the communication is going on. The time between two

consecutive speed and direction changes follows an exponential law; once the call ends up; the voice user exits the system. A distance between base stations of 3 kilometers has been considered. MS and traffic consequently, are uniformly spread over the simulation layout. Most parameters have been borrowed from similar studies and handbooks [12-16]. As to power control, handoff management and call access control algorithms refer to descriptions in [7].

TABLE II. OFFERED TRAFFIC AND INTER ARRIVAL TIME

$\overline{\eta_{DL}}$	$1/\lambda_t$
50%	0.090 s
70%	0.064 s
90%	0.050 s

3. Variables Under Study

The first group of variables studied in this work aims at providing information about the QoS offered by the system. Two QoS figures have been considered in this study: the call blocking probability (*BP*) and the probability of forced termination of a call, i.e. due to handoff failure (*FP*). Both magnitudes are clearly perceived by the user and can be measured at the user plane instead of other magnitudes such as the handoff dropping probability that is only perceived at the network plane [17].

Assessing the traffic load at peak hours helps dimension system elements (mainly the radio resources and fixed equipment). These elements require substantial investment. As the resources are expensive and have to be kept to a minimum, dimensioning has to assume that the demand for a low portion of users cannot be met when system is heavily loaded. *BP* is the probability that a fresh call cannot be accepted due to the lack of available resources in the set of Base Stations (BS) that should be able to serve the call. In the simulator, this output is computed as:

$$BP = \frac{Number\ of\ Blocked\ Fresh\ Calls}{Number\ of\ Fresh\ Call\ Attempts}. \quad (1)$$

FP is the forced termination probability, i.e. the probability that an ongoing call is interrupted due to a HO failure. The handoff failure occurs every time that a call looses coverage from the serving BS without having seized a channel in a new one. The simulator obtains this output as:

$$FP = \frac{Number\ of\ Interrupted\ Calls}{Number\ of\ Established\ Calls}. \quad (2)$$

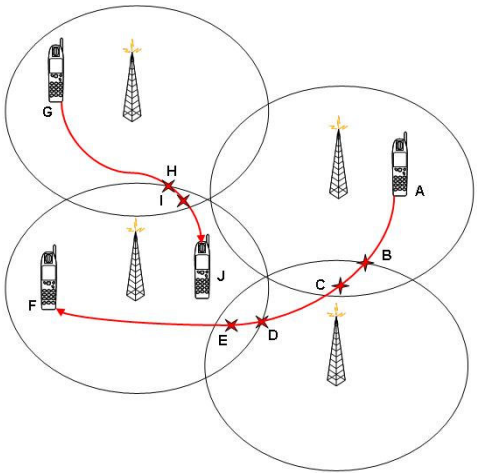


Fig. 1. Teletraffic variables

The second group of variables considered in the study is composed by three Teletraffic variables whose definitions follow:

1) Channel holding time. Gathers the time periods in which a channel is allocated to a call, starting the call as a fresh one (i.e. a new originated call) or as an accepted *handoff*. Fig. 1 shows a scenario with two Mobile Stations (MSs) setting up two different calls. The letters indicate the time marks and the lines trace the geographical path followed by each MS while the call is ongoing. Under these conditions, channel holding time samples include (A - C), (C - E), (E - F), (G - I) and (I - J) time values.

2) Time between two consecutive handoff arrivals. Stores the time elapsed between two consecutive *handoff* arrivals at the same cell. This variable only considers the first request made by each *handoff* process received in the cell, i.e., the reattempts associated to *handoff* processes that were not immediately served are not taken into account. These samples include the (D - H) value, under the assumption that attempt D took place before H.

3) Handoff duration. Saves the time elapsed since an MS requests a handoff until the handoff is supplied or dropped. In Figure 1, this sample includes the (B - C), (D - E) and (H - I) values.

4. Analysis of QoS Results

Results for QoS variables concerning comparison between exponential and lognormal service time are shown in Table III. For those simulations, the distance between BSs is

3000 m. The ratio of around 1 to 2 handoffs per call could have been roughly estimated as follows: considering inter-cell distance and the presence of 3 sectors per cell, 1000 m. is a rough estimation of the length that a MS runs within a sector. At 13.9 m/s the time within a sector is around 72 s. and a call that takes 120 s. involves 120/72=2 sectors, hence 1 handoffs. The random nature of the radio path including shadowing increases this ratio.

Notice how in Table III a good balance is maintained between the blocking and forced termination probability. The impact of increasing the load is more noticeable on the latter while it is always kept to low percentages. This is desirable and is achieved by means of the power control and CAC that favors continuity of service and thus decreases the interruption probability to the detriment of the call blocking probability. If the bound for the CAC changed from 1 dB to 2 dB, one should expect lower blocking probabilities but more interrupted calls. While the main objective of the CAC is to avoid the degradation of the quality of the conversation, a side impact is the desirable control of the balance between *BP* and *FP*.

TABLE III. RESULTS OF THE QOS VARIABLES

Load	**Exponential**			**SCV 1 (Log)**		
	HO ratio	**BP (%)**	**FP (%)**	**HO ratio**	**BP (%)**	**FP (%)**
50 %	1.15	8.05	0.05	1.15	8.12	0.01
70 %	1.60	14.95	0.70	1.70	15.16	0.13
90 %	1.81	21.25	1.70	1.83	21.78	0.86
Load	**SCV 3 (Log)**			**SCV 10 (Log)**		
	HO ratio	**BP (%)**	**FP (%)**	**HO ratio**	**BP (%)**	**FP (%)**
50 %	1.16	8.45	0.02	1.22	12.32	0.03
70 %	2.24	18.78	0.12	1.93	23.13	0.65
90 %	1.86	22.12	0.98	2.36	26.71	2.13

A difference between exponential service time and lognormal one, both with the same average duration and square coefficient of variation (SCV) from 1 to 10, must be pointed out. Handoff to fresh-call ratio is smaller in exponential case as can be seen from Table III. This fact can be explained considering that the exponential model overestimates the presence of shorter calls that cannot generate a high number of handoff s because they do not have the time to cross an entire sector.

The impact of this higher handoff ratio should be taken into account when dimensioning the capacity for handoffs access and the signaling network. On the other hand, the consequences of the lognormal distribution on the blocking and interruption probabilities are not so clear. For heavy load, performance is poorer with the lognormal (notice that the user perceives *FP* as a worse service degradation than *BP*). But for medium and light loads performance is better for the lognormal case.

A different behavior can be observed comparing the results in Table III for SCV of 3 and SCV of 10, in comparison with exponential service time. Such a difference can be attributed to the increased amount of large call due to the variation of second moment parameter of service time used in simulations (a squared coefficient of variation of 3 and 10 in lognormal case). A difference between SCV of 3 and SCV of 10, both with the same average duration must be pointed out. As can be seen from Table III, both BP and FP increase if traffic load increases or if a larger service time SCV is considered. Such reaction is expected if increasing in traffic load is taken into account, but it is clearly magnified by the larger SCV considered in simulations. This fact can be explained considering that a larger SCV overestimates the presence of larger calls that generate a high number of HOs and so the probability for a call to be dropped is higher. In this case, the effect of increasing SCV is more visible in BP because of the presence of a CAC that always protects the established call avoiding a new call to enter the system and so decreasing the quality of an ongoing call.

The impact of this higher BP and FP probability should be taken into account when dimensioning a network in prevision of increasing of larger calls or connections. The consequences of the increase in SCV of lognormal service time distribution on the blocking and interruption probabilities are clear: for medium and heavy loads, a better balance between BP and FP has to be achieved. According to authors' opinion, such balance can be reached considering a combination of admission control and handoff management algorithms.

5. Analysis of Teletraffic Results

5.1. Channel Holding Time

For lognormal call duration the channel holding time average is lower than the one for the exponential call duration as shown in Table IV; this is in agreement with the higher handoff ratio associated to lognormal service time distribution reported in Table III. Simulation results for this variable have a *SCV* lower than one. For both lognormal and exponential call duration the channel holding time average is longer than the rough estimation at the beginning of Section 4. It is important to notice that as traffic increases, channel holding time decreases due to the higher *FP* probability.

About average channel holding time, looking at Table IV, we can notice that incrementing SCV increases the average connection time to a channel: a call will spend more time occupying the same channel. A clear dependence between SCV of a call and connection duration can be pointed out. In this case load is a variable that seems not to have a relevant influence, as the effect of increasing load is quite subtle on SCV of channel holding time.

TABLE IV. CHANNEL HOLDING TIME RESULTS

Load	Exponential		SCV 1 (Log)		SCV 3 (Log)		SCV 10 (Log)	
	Average	SCV	Average	SCV	Average	SCV	Average	SCV
50%	72.29	0.05	72.49	1.05	73.33	1.8	87.29	2.05
70%	55.95	0.7	56.14	1.21	57.96	1.93	68.85	2.23
90%	53.25	1.03	53.92	1.52	55.96	1.98	63.15	2.26

5.2. Handoff Traffic

This random variable represents the time between two consecutive handoffs to the same cell (i.e. sector in a BS). In this case two parallel studies have been carried out: one including and the other neglecting simultaneous handoff requests. It must be noticed that, in the simulated scenarios, simultaneous handoff requests are highly probable events since the high user mobility and traffic increase leads to a high rate of handoff s per call. Thus, the probability density function has a peak at the origin, due to the high frequency of simultaneous handoffs. For fitting purposes it is easier to fit separately the simultaneous and non-simultaneous handoffs than trying to fit a distribution for all. As a consequence, the characterization of this random variable was conducted in two steps. First, the probability of two or more handoff requests arriving simultaneously at the same cell is calculated. Simultaneous handoffs will thus be eliminated from the empirical sample prior to the second step, which consists of calculating the optimal parameters of the fitting distribution after having discarded the zero values. The combination of these two distributions completely defines the variable under study.

TABLE V. RESULTS OF HANDOFF TRAFFIC

Load	Exponential			SCV 1 (Log)			SCV 3 (Log)			SCV 10 (Log)		
	Avg_1	Avg_2	SCV	Avg_1	Avg_2	SCV	Avg_1	Avg_2	SCV	Avg_1	Avg_2	SCV
50%	6.34	16.71	0.29	6.12	16.45	0.18	5.44	14.79	0.18	3.33	10.71	0.29
70%	3.86	10.95	0.14	3.54	10.56	0.2	2.58	8.11	0.38	1.86	7.13	0.52
90%	3.44	8.23	0.11	3.37	7.98	0.19	2.21	7.68	0.4	1.58	6.43	0.58

Table V shows the achieved results, where Avg_1 represents the average of the variable including simultaneous handoffs while Avg_2 and SCV represent the average and the squared coefficient of variation of the variable not including simultaneous handoffs. Notice how the average time between handoffs decreases when the load increases: this was

expected since more load involves more ongoing calls, hence more handoff attempts received by each cell.

Considering handoff traffic, the impact of increasing SCV is reflected in higher handoff rates, both for high and very high loads. A direct consequence of increasing handoff rate is the raise of handoff traffic and probability for a call to be dropped. Such probability in this case remains between desirable thresholds due to the presence of a CAC that protects ongoing calls at cost of a higher blocking probability.

5.3. Handoff Duration

In characterizing this variable, it must be taken into account that, if a system is properly dimensioned, the probability of immediately obtaining a new channel (i.e. at the first handoff attempt) is high. This is due to the fact that, since the CAC only applies to fresh call arrivals and fresh calls do not automatically reattempt in a very short period, handoffs have a higher priority in the competition for available channels. Thus, the probability density function for this variable is given by a delta function at zero delay, plus a probability density function that only considers delays longer than zero. In this work, these two components were separately studied: we considered separately the probability of immediately serving a handoff and the characterization of the delayed handoffs.

TABLE VI. RESULTS FOR HANDOFF DURATION

Load	**Exponential**			**SCV 1 (Log)**		
	Avg	**SCV**	**P_0**	**Avg**	**SCV**	**P_0**
50 %	0.2632	0.191	85.902	0.1134	0.123	99.079
70 %	0.5612	0.224	74.735	0.1186	0.125	98.698
90 %	0.9635	0.257	65.072	0.1232	0.127	97.979
Load	**SCV 3 (Log)**			**SCV 10 (Log)**		
	Avg	**SCV**	**P_0**	**Avg**	**SCV**	**P_0**
50 %	0.1217	0.156	98.204	0.2160	0.142	93.356
70 %	0.1381	0.158	96.885	0.2254	0.184	93.023
90 %	0.1543	0.160	94.345	0.2456	0.203	92.679

Table VI shows the results for the handoff duration. In the table, average and SCV values are calculated considering handoffs that are immediately accepted. Parameter P_0 refers to the probability for a handoff call of being immediately served.

Note that the delay increases while the probability of immediate handoff decreases: this impact increases along with the offered load. *Average* is smaller for a SCV value of 3 than

for a SCV value of 10 both for medium and high loads while the probability that a call can be immediately served is also smaller. This is conjectured to depend from the presence of longer calls that using for more time the resources of a cell avoid that incoming calls from other cell obtain a channel immediately.

6. Conclusion

In this work, simulations results have been analyzed in order to model UMTS cellular networks in suburban and rural scenarios. The study of the QoS variables shows that the CAC allows a good control of the balance between the blocking and interruption probabilities, balance that is not maintained in rural scenario, where the presence of CAC generates a higher number of call that cannot access the system.

The impact of a lognormal distributed call duration (instead of the classical exponential) can be summarized in a larger handoff to fresh call ratio, thus causing a shorter channel holding time and a higher FP probability for high load handoff duration is smaller. As a consequence, performance is different for lognormal distributions especially for high and very high loads. Some behavior's differences can be also noticed between exponential and lognormal with SCV of 1. Therefore, networks planners must carefully consider the actual call duration distribution, in a way to avoid under sizing the system.

The impact of increasing SCV of service time (starting from 1 to 3 and 10), can be summarized into higher BP and FP, even if the presence of a restrictive admission control always protects ongoing call at cost of new calls that try to access the system.

Considering Teletraffic variables, the impact can be observed in higher average connection times and the SCV of connection times seem to follow the behavior of SCV of service time. Another important result is the increasing of handoff rates in relation to increasing in SCV of service time. This behavior has to be considered in planning a network, to manage the impact of increasing in signalization connected to increasing handoffs.

For channel holding time, the impact of the lognormal call distribution is a reduction of the average channel holding time in agreement with the larger handoff ratio. Regarding the time between two consecutive handoffs to the same cell is higher handoff rates and, as a consequence, the raise of handoff traffic and probability for a call to be dropped. Handoff duration also increase with the offered load.

References

1. D. Hong, S.S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures", IEEE TVT. 35(3), pp. 77-92, August 1986.

2. Y. Fang, “Hyper-Erlang Distribution and It’s Applications in Wireless and Mobile Networks”, Wireless Networks (WINET), Vol. 7, No. 3, pp. 211-219, May 2001.
3. E. Chlebus, T. Zbiezek, “A novel approach to simulation of mobile networks”, 12th ITC Specialist Seminar on Mobile Systems and Mobility, pp. 261-274, March 2000.
4. F. Barceló, J. Jordán, “Channel Holding Time Distribution in Public Telephony Systems (PAMR and PCS)”, IEEE Trans. Veh. Tech., Vol. 29 No. 5, pp. 1615-1625, September 2000.
5. F. Barceló, J. I. Sánchez, “Probability distribution of the Inter-Arrival time to Cellular Telephony Channels”, IEEE 49th Vehicular Technology Conference (VTC Spring), pp. 762-766., 1999
6. M. Ruggieri, F. Graziosi, F. Santucci, “Modeling of the Handoff Dwell Time in Cellular Mobile Communications Systems”, IEEE Trans. Veh. Tech., Vol. 47, No 2, pp. 489-498, May 1998.
7. A. Spedalieri, I. Martin-Escalona, F. Barceló; “Simulation of Teletraffic variables in UMTS networks: impact of lognormal distributed call duration”, Wireless Communications and Networking Conference, 2005 IEEE, 13-17 March 2005 Vol. 4 Page(s):2381 - 2386.
8. V. Pla y Casares-Giner, V., “Analytical-numerical study of the handoff area sojourn time”, IEEE GLOBECOM 2002, Vol.1, pp. 886 – 890, 17-21 Nov.2002.
9. E. Chlebus, “Empirical validation of call holding time distribution in cellular communications systems”, Proc. 15th International Teletraffic Congress, Elsevier Science B. V. , pp. 1179 – 1189, 1997
10. V. A. Bolotin, “Modelling Call Holding Time Distributions for CCS Network Design and Performance Analysis”, IEEE Journal On Selected Areas in Communications, Vol. 12, No. 3, pp. 433- 438, April 1994.
11. A.J. Viterbi, “The Orthogonal – Random Waveform Dichotomy for Digital Mobile Communications”, IEEE Personal Communications, Vol 1, Is 1, pp. 18 -24, 1994.
12. H. Holma y A. Toskala, WCDMA for UMTS, John Wiley & Sons, 2000.
13. ITU-D, Handbook “Teletraffic Engineering”, January 2005.
14. 3GPP Technical Specification 25.101, UE Radio Transmission and Reception (FDD).
15. 3GPP Technical Report 25.942, RF System Scenarios.
16. L. Nuaymi, P. Godlewski, X. Lagrange, “Power allocation and control for the downlink in cellular CDMA networks”, 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol 1, pp. C-29 - C-33, 30 Oct. 2001.
17. F. Barcelo, “Performance Analysis of Handoff Resource Allocation Strategies through the State-Dependent Rejection Scheme”, IEEE transactions on Wireless Communications, Vol. 3, Issue 3, pp. 900- 909, May 2004.

Delivering Adaptive Scalable Video over the Wireless Internet

Pavlos Antoniou, Vasos Vassiliou, and Andreas Pitsillides

Networks Research Laboratory, Computer Science Department, University of Cyprus*
{paul.antoniou,vasosv}@cs.ucy.ac.cy,
andreas.pitsillides@ucy.ac.cy

Abstract. A large portion of the emerging and future wireless Internet traffic is foreseen to be consumed by video streams. However, delivering video over wireless networks poses a lot of challenges. Network congestion and wireless channel errors yield tremendous packet loss leading to degraded video quality. One of the most critical issues for video applications is to ensure that the quality of service (QoS) requirement will be maintained at an acceptable level, providing responsiveness to the time-variant network conditions as well as scalability and fairness among concurrent users. In this paper, we study the performance of a novel fuzzy-based adaptive mechanism which takes into account a combination of Network Adaptation Techniques with Content Adaptation Techniques in order to achieve graceful performance degradation when network load increases and network conditions deteriorate. Our performance evaluations indicate that our approach finely adapts the video stream bit rate to the available bandwidth, maintains responsiveness to dynamic changes and achieves scalability and fairness as well as high and stable objective quality of service.

1 Introduction

The overwhelming majority of today's handheld devices like mobile phones, PDAs and laptops are capable of streaming video content. Therefore, video transmission over the Internet is considered to be the prime candidate for being the next killer application.

Needless to say that video communications face a lot of challenges. Compressed video streams (like MPEG) exhibit large variations in their data rates something which makes their management in a packet-based best-effort network like IP extremely difficult. Moreover, the unpredictable nature of various heterogeneous networks within the Internet primarily in terms of bandwidth, latency and loss variation make the transmission of the compressed video streams an even more challenging task. The problem is worsened when we consider mobile users connecting with wireless terminals due to the erroneous and time-variant conditions of the wireless environment.

Under these circumstances, video transmission applications need to be responsive to dynamic changes and different demands. Thus, they need to implement highly scalable and adaptive techniques in terms of content encoding and transmission rates in order to cope with the increased network heterogeneity and complexity. Towards this direction,

* This work was partly funded by the UCY ADAVIDEO project.

the combination of Content Adaptation Techniques (CATs) with Network Adaptation Techniques (NATs) is considered to be an imperative need. CATs deal with adaptation of the video content to the desirable transmission rate using primarily scalable video approaches. Scalable video approaches can solve the variable bandwidth problem only if the streaming architecture is able to track the available bandwidth and react without latency. Thus, we consider NATs which deal with the end-to-end adaptation of real time video application needs to the network parameters using algorithms which take into account the state and/or load of the network and the type of errors.

In this paper, we present a fuzzy-based approach for the adaptive delivery of video streams under variable connection characteristics, which is targeted for video delivery in wireless and mobile environments. Our approach involves a new feedback mechanism that works in conjunction with a fuzzy decision algorithm. We study the performance of our approach with respect to responsiveness to dynamic changes, graceful co-existence with cross traffic, scalability and fairness among concurrent mobile and wireless users.

The remainder of this paper is organized as follows: Section 2 presents and analyzes the architecture of the adaptive mechanism. Section 3 deals with the evaluation setup and scenarios. Section 4 presents some performance results. Section 5 concludes the paper and discusses future work.

2 Adaptive Video Streaming Components

Our approach consists of two basic components, namely a feedback mechanism and a fuzzy-oriented decision algorithm depicted in Fig. 1. The feedback mechanism combines receiver's critical information on the perceived quality as well as measurements obtained by the core network in order to evaluate the available bandwidth of the network path. The estimated available bandwidth is then fed into the decision algorithm which decides in a fuzzy manner the optimal number of layers that should be sent by adding or dropping layers.

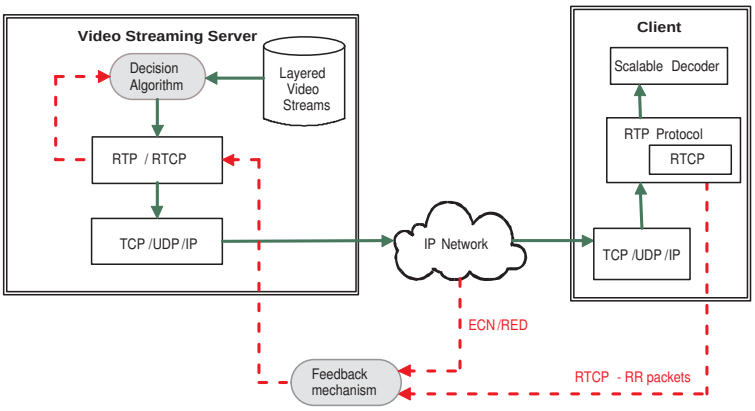


Fig. 1. Adaptive fuzzy-based video streaming architecture.

Fig. 1 illustrates the architecture of a unicast-oriented system using our adaptive fuzzy-based approach. The two outlined components focus on the adaptation of the layered video content to the available network bandwidth. We assume that each video stream is encoded in multiple layers stored at the sender side. The layered video content is transmitted over an RTP/RTCP connection [1]. Dashed arrows track the path of control packets whereas solid arrows track the path of video data packets.

The role of the feedback and adaptation components is to link the quality demand of video-enabled applications to the underlying network. Network adaptation is assisted by a proper content adaptation technique which is carried out by layered video encoding.

2.1 Feedback Mechanism

The feedback mechanism collects QoS information (e.g. loss rate, jitter) from both the core network and the receiver that will be used for the evaluation of the available bandwidth of the path between the sender and a receiver.

Each receiver sends reception statistics using dedicated RTCP packets called Receiver Report (RR) packets which carry reception statistics. Among them, the packet loss fraction within an interval is given by the number of packets expected divided by the number of lost packets during the interval. The loss rate per second (LRPS) can be obtained by dividing the loss fraction by the difference in RRs timestamps. The difference between two successive values of LRPS can be used in order to track the increasing or decreasing trend of packet loss percentage.

Additionally, network elements as, for example, routers within the network path may explicitly notify the sender about the current status of congestion within the core network. These notifications can be efficiently used for the evaluation of the available bandwidth. The Explicit Congestion Notification (ECN) mechanism mentioned in [2] is used for the notification of congestion to the end nodes in order to prevent unnecessary packet drops. ECN option allows active queue management (AQM) mechanisms such as, for example RED [3] or Fuzzy-RED [4] to probabilistically mark packets. The number of marked packets within a given period may provide a meaningful reference about the congestion status. The receiver collects these data and sends them back to the sender using a dedicated field of the RR packets.

2.2 Fuzzy Decision Mechanism

The decision algorithm which is implemented at the sender side, processes the feedback information and decides the optimum number of layers that will be sent using fuzzy logic control. Our fuzzy decision algorithm is based on two linguistic input variables and one output variable. All quantities in our system are considered at the discrete instant kT, with T the decision period.

Our first linguistic input variable involves the LRPS parameter. LRPS(kT) is the loss rate per second at each decision period and LRPS(kT−T) is the loss rate per second with a delay T. The linguistic variable D_{LRPS}(kT) gives the increasing or decreasing trend of the LRPS and can be evaluated by:

$$D_{LRPS}(kT) = LRPS(kT) - LRPS(kT - T) \tag{1}$$

The LRPS parameter is lower and upper bounded by 0 and 1 respectively. Thus, the D_{LRPS}(kT) parameter ranges from -1 to $+1$.

For the second input linguistic variable we use the number of packets that have the ECN bit set within a period. The receiver calculates periodically this number called N_{ECN}(kT) and send it back using a dedicated field of the RR packet. The sender extracts this value and calculates a scaled parameter, N_{ECNsc}(kT), which ranges from -1 to $+1$, and represents the percentage of packets marked within this period. Eq. 2 is used to obtain the scaled parameter N_{ECNsc}(kT):

$$N_{ECNsc}(kT) = \frac{N_{ECN}(kT)}{N_{ps}(kT)}, \tag{2}$$

where N_{ps}(kT) is the number of packets sent within the same period. Therefore, we calculate the parameter DN_{ECNsc}(kT), which gives the increasing or decreasing trend of the number of marked packets. The DN_{ECNsc}(kT) is upper and lower bounded by $+1$ and -1 respectively, and can be evaluated by:

$$DN_{ECNsc}(kT) = N_{ECNsc}(kT) - N_{ECNsc}(kT - T) \tag{3}$$

Our fuzzy system [5] processes the two linguistic input variables based on the predefined if-then rule statements (rule base) shown in Table 1, and derives the linguistic output variable a(kT), which is defined for every possible combination of inputs. The defuzzified crisp values of a(kT) can be used by the decision algorithm for the evaluation of the available bandwidth using the formula:

$$avail\_bw(kT) = a(kT) * avail\_bw(kT - T) \tag{4}$$

The defuzzified output value is selected to range from 0.5 to 1.5. Thus a 'gradual' increase is allowed when there is available bandwidth and reduced congestion, whereas quick action is taken to reduce the rate to half in case of severe congestion.

Table 1. Linguistic Rules[1].

a(kT)		DN_{ECNsc}(kT)						
		NVB	NB	NS	Z	PS	PB	PVB
D_{LRPS}(kT)	NVB	H	H	B	B	Z	S	VS
	NB	H	VB	Z	Z	Z	S	VS
	NS	B	Z	B	Z	Z	S	VS
	Z	B	Z	Z	B	Z	S	VS
	PS	Z	Z	Z	Z	S	S	VS
	PB	Z	Z	Z	Z	S	S	VS
	PVB	S	S	S	S	VS	VS	VS

[1] Table Content Notations: Negative/Positive Very Big (NVB, PVB), Negative/Positive Big (NB, PB), Negative/Positive Small (NS, PS), Zero (Z), Very Small/Big (VS, VB), Small/Big (S, B), Medium (M), Huge (H).

Our decision algorithm has to decide which layers should be sent according to the available bandwidth, based on a non aggressive layer selection approach. The server hosts an appropriate number of layers which correspond to different transmission rates. To avoid ping-pong effects there should not be a transition to an upper level layer every time the available bandwidth exceeds the threshold of a specific rate that corresponds to a higher layer. Instead, a time hysteresis is introduced in order to avoid frequent transitions from one layer to another. More detailed description can be found in [5].

3 Evaluation Setup and Scenarios

3.1 Topology

Fig. 2 illustrates the dumbbell topology we used for the performance evaluation of our approach. A bottleneck link was simulated using two routers directly connected with a link having variable characteristics. All the other wired links have constant bandwidth (10Mbps) and propagation delay (1ms). A video streaming server is attached to the first router. Mobile clients are wirelessly connected to an access point which is attached to the second router. In order to make our scenarios more realistic we added FTP and web-like cross traffic initiated by the FTP server and the WEB server which are both connected to the first router. Wired clients were used to initiate cross traffic.

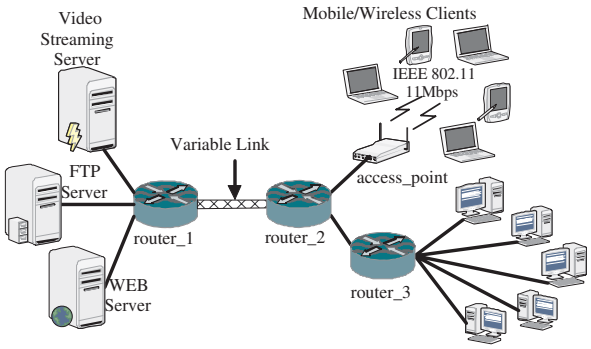


Fig. 2. Evaluation topology.

3.2 Variable Test Parameters

The different parameter values used to characterize the variable link between the routers are shown in Table 2. The bandwidth of the variable link ranges from 64Kbps to 4Mbps, while the propagation delay varies from 10ms to 800ms. The choice of the parameters used in the video quality evaluations is based on the representative characteristics of wired and wireless networks. For example, the link bandwidth can be considered as either the last hop access link bandwidth or the available bandwidth to the user. The values chosen can represent typical wired home access rates (e.g. modem, ISDN, xDSL).

The maximum buffer capacity was set to 50 packets, and RED parameters as shown: $(min_{th}, max_{th}, p_{max}) = (10, 30, 0.1)$. Moreover the interval T between transmissions of RR packets (decision period) was set to 0.5 seconds. The selection of 0.5 seconds is dictated by the desire to maintain responsiveness to changes in the network state.

3.3 Test Sequences

The video sequence used in this study was the well known real test video named *Foreman*, which is a stream with a fair amount of movement and change of background. The sequence has temporal resolution 30 frames per second, GoP (Group of Pictures) pattern IBBPBBPBBPBB, and spatial resolution 176x144. We encoded this sequence using a publicly available MPEG4 encoder [6] in 8 different bit rates as shown in Table 2. Each encoded video stream corresponds to a separate layer. Since the encoding of the sample video sequences is based on MPEG4, individual frames have variable sizes.

Table 2. Variable Link and Video Parameters.

Video Stream Bit Rate	*Link Bandwidth*	*Propagation Delay*
64 Kbps 384 Kbps	64 Kbps 768Kbps	10 ms
96 Kbps 512 Kbps	128 Kbps 1 Mbps	100 ms
128 Kbps 768 Kbps	256 Kbps 2 Mbps	200 ms
192 Kbps	384 Kbps 4 Mbps	400 ms
256 Kbps	512 Kbps	800 ms

3.4 Data Collection

All the aforementioned experiments were conducted with an open source network simulator tool ns2 [7]. Due to the inadequacy of the existing ns2 modules, we implemented some new software modules [5]. Based on the open source framework called EvalVid [8] we were able to collect all the necessary information needed for the objective video quality evaluation like PSNR values. Video quality is measured by taking the average Peak Signal-to-Noise Ratio (PSNR) over all the decoded frames. Some new functionalities were implemented in ns2 from [9] in order to support EvalVid.

4 Results

In this section we present and investigate the performance of our approach based on the results obtained from the above scenario evaluations. The time varying behavior of the network environment is carried out through cross traffic patterns. Section 4.1 studies the responsiveness of the proposed approach to dynamic changes of the network environment. Section 4.2 investigates the effect of link bandwidth and propagation delay on the received video stream quality in terms of PSNR. Section 4.3 deals with scalability

and fairness issues. Finally Section 4.4 focuses on the system capacity with respect to the number of users that can be supported by a video streaming server.

Objective quality metrics like, PSNR, cannot characterize fully the response and the end satisfaction of the viewer. Subjective quality assessment is more a reliable method, as the perceived measure of the quality of a video is done through the human "grading" of streams which helps collect and utilize the general user view (Mean Opinion Score, MOS). To this end, the relationship between the MOS and the PSNR, based on the same *Foreman* video sequence, in a similar network environment is demonstrated in [10].

4.1 Responsiveness to Dynamic Network Changes

We investigate the ability of the fuzzy rate controller to sense the available bandwidth of a bottleneck link in the presence of various cross traffic patterns, and adapt the transmission rate of a 1Mbps scalable CBR non trace-based video stream. The video stream is transmitted over a bottleneck link having constant bandwidth of 1Mbps to a mobile user. We consider three kinds of traffic patterns, namely, (a) multiple CBR connections which are superimposed progressively, (b) FTP traffic, and (c) Web-like traffic.

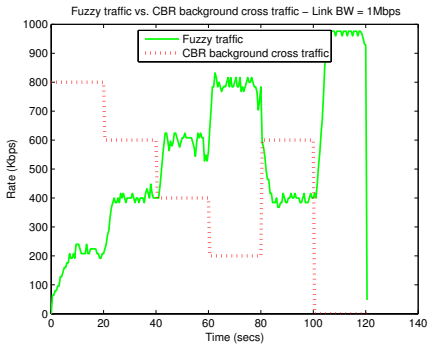
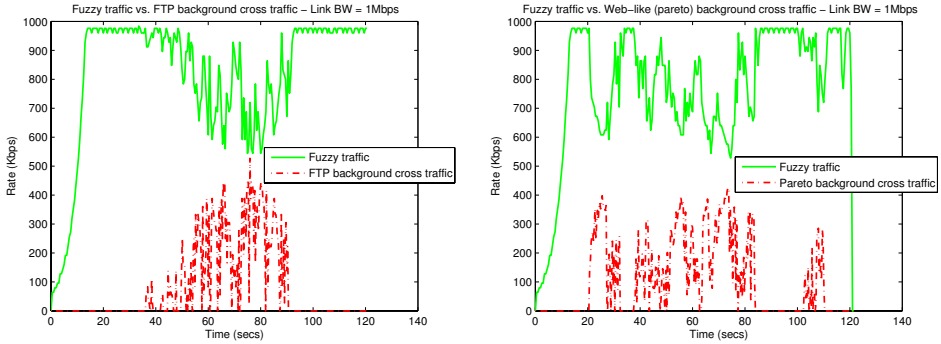


Fig. 3. Instantaneous rate for 1Mbps bottleneck link with CBR cross traffic.

Fig. 3 depicts the instantaneous transmission rate of the layered CBR video stream as the CBR traffic rate changes over the time. The CBR cross traffic rate ranges from 200Kbps to 800Kbps. As can be seen, the video transmission rate driven by the fuzzy rate controller, evolves at a slow and smooth pace in order to respond to the network and quality conditions, but also prevent unnecessarily many fluctuations.

Fig. 4(a) shows the transmission rate of the layered CBR video stream in the presence of FTP traffic. Although the FTP cross traffic is more bursty than CBR shown in Fig. 3, the fuzzy controller senses the available capacity of the bottleneck link and finely adapts the video rate to it. The fuzzy-controlled flow appears to be TCP-friendly against an FTP flow, as it does not aggressively consume the available bandwidth.

Fig. 4(b) depicts the instantaneous transmission rate of the layered CBR video stream in the presence of web-like cross traffic. We simulated web-like traffic using



(a) FTP cross traffic. (b) Web-like (pareto) cross traffic.

Fig. 4. Instantaneous rate for 1Mbps bottleneck link with FTP/Web-like cross traffic.

a single ON/OFF UDP source, with ON and OFF times drawn from a Pareto distribution. The mean ON time is 350ms, the mean OFF time is 650ms, and during ON time, the UDP sending rate is 400Kbps. The shape parameter of the Pareto distribution is set to 1.11. Even though the web-like traffic is extremely bursty and unpredictable, the fuzzy controlled flow maintains responsiveness during heavy and time-variant loads.

4.2 Effect of Link Bandwidth and Propagation Delay on the QoS

In order to study the effect of link bandwidth and propagation delay on the received QoS, we conducted scenarios involving one wireless mobile user that receives streaming video over the topology shown in Fig. 2 in the absence and presence of cross traffic.

Fig. 5 reveals that in the absence of cross traffic, the PSNR values are increasing at a steady pace (up to 36.5dB) as the link bandwidth increases. PSNR values are significantly lower (less than 20dB) in scenarios where the link bandwidth is equal to the bit rate of the lowest layer (64Kbps), since there is a strong possibility of packet loss. In high bandwidth links (above 512Kbps), the PSNR values are slightly higher for low delay values. On the other hand, in medium bandwidth links (between 256Kbps and 512Kbps), the PSNR values are slightly lower for low delay values. This observation is attributed to the fact that the longer the propagation delay the longer the interval between reception of two successive RR packets. Under these circumstances, the system will experience delayed decision-making that will influence the quality of the video stream. If the link bandwidth is high enough to sustain the video transmission rate, a delayed decision will result to smaller PSNR values because the content adaptation evolves at a slow pace. In the contrary, low delay values will result to higher PSNR since the content adaptation to network parameters evolves at a faster pace. In the case of a low bandwidth link, delayed decisions will benefit the system since the sending rate will be kept in lower levels. This results to higher PSNR values due to the small number of packets lost, since rapid changes in the number of layers sent are avoided.

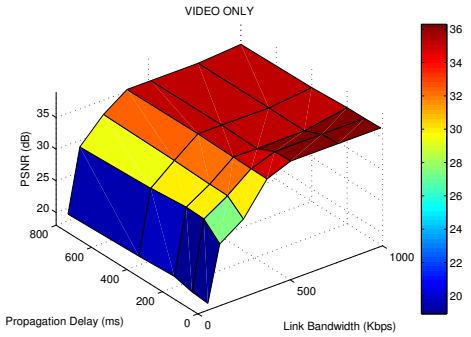
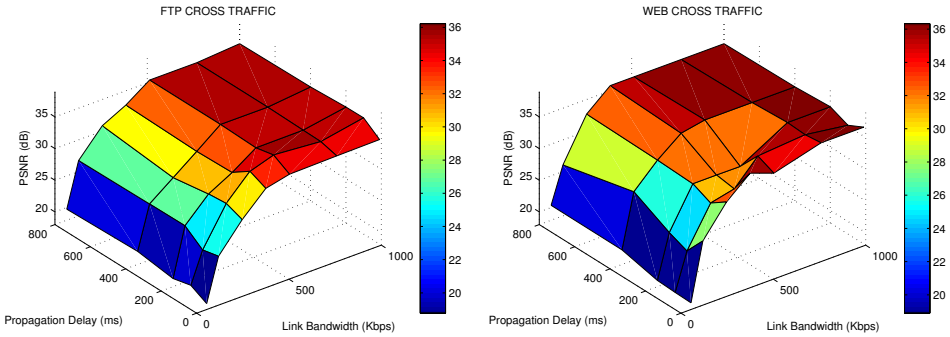


Fig. 5. Mean PSNR vs. Link BW and Prop. Delay.

Fig. 6(a) shows PSNR for scenarios involving FTP cross traffic. We observe a slight decrease in PSNR for scenarios having link bandwidth less or equal to 256Kbps due to the excessive FTP traffic load. As the link bandwidth increases (more than 256Kbps), the quality of a video stream is not severely affected by the FTP traffic since the decision algorithm adjusts the number of layers sent, according to the variable network conditions. Moreover, we perceive a lower objective quality for low propagation delay values, because the FTP rate evolves at a faster and more aggressive pace than in scenarios with longer delay due to the inherent characteristics of the underlying TCP protocol, resulting in high packet drop rates.



(a) FTP cross traffic. (b) Web cross traffic.

Fig. 6. Mean PSNR vs. Link BW and Prop. Delay, with FTP/Web cross traffic.

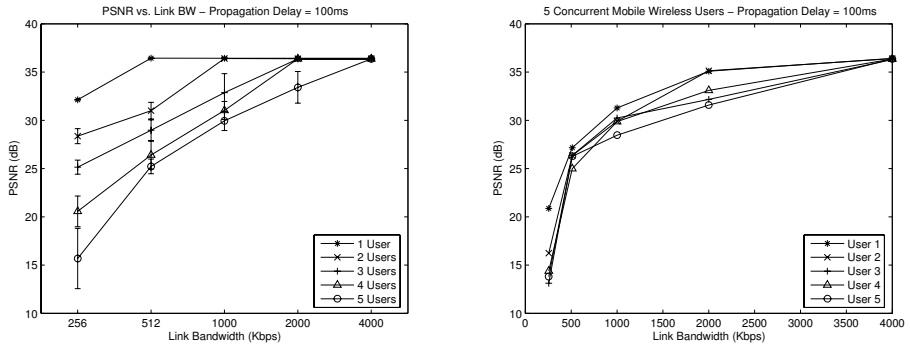
The effect of propagation delay and link bandwidth on the PSNR in the presence of web traffic is presented in Fig. 6(b). Five fixed users (see Fig. 2) are used to simulate web traffic. Each user initiates two sessions and each session consists of 10 web pages.

Session 1 has the following characteristics: exponential inter-page interval (mean = 200ms), Pareto II web page size (mean = 4 objects, shape = 1.5), exponential inter-object interval (mean = 10ms), and Pareto II object size (mean = 4 packets, shape = 1.2). In addition, Session 2 has the following characteristics: exponential inter-page interval (mean = 300ms), constant web page size (1 object), exponential inter-object interval (mean = 10ms), and Pareto II object size (mean = 10 packets, shape = 1.2). As can be seen, the shape of the quality surface obtained from these scenarios is somehow similar to this concerning FTP traffic (Fig. 6(a)). The quality of the received video stream seems to deteriorate more than in FTP for low propagation delay values when link bandwidth ranges from 64Kbps to 768Kbps. As mentioned in the case of FTP traffic, this is justified by the aggressive behavior of the TCP protocol on which the web traffic is based, as well as by the aggressive characteristics (small intervals between web pages and embedded objects) of the web traffic.

The aforementioned scenarios reveal that our approach can finely adapt the video stream bit rate to the available bandwidth. Based on subjective evaluations presented in [10], the *Good* and *Excellent* categories of MOS define the lowest limit for acceptable objective quality, which is 27dB. Thus, our results demonstrate that our system provides high objective quality (above 27dB) both in the absence and in the presence of cross traffic.

4.3 Scalability and Fairness in Multiple Concurrent Mobile/Wireless Users Scenarios

We investigate the ability of our unicast-oriented system to provide scalability and fairness, taking into account that the decision algorithm operates individually for each user. Our scenarios involve multiple concurrent wireless and mobile users, having the same characteristics and requirements. Fig. 7(a) depicts the mean PSNR between all users in each scenario, for scenarios involving one, two, three, four, and five users, when the propagation delay over the bottleneck link is 100ms.



(a) Scalability in scenarios with multiple users. (b) Fairness in scenario with 5 concurrent users.

Fig. 7. Scalability and fairness concerning users with the same characteristics/requirements.

As can be seen, our system achieves scalability by sharing the available bandwidth to all active users, even in the cases where the link bandwidth is not high enough to sustain the aggregated video transmission rate. As the number of concurrent users scales up, more users can be supported by diminishing the received quality per user thus offering graceful degradation. Similarly, fairness is achieved when link bandwidth is inadequate of handling aggregated traffic. Fig. 7(b) shows that in the case of 5 concurrent users, the available bandwidth is fairly shared among them as they receive almost the same quality in terms of PSNR.

4.4 System Capacity

Fig. 8 provides an intuition for the capacity of the system with respect to the number of wireless and mobile users that can be supported by a video streaming server, taking into account the bottleneck link bandwidth. The diagram depicts the mean quality of service in terms of PSNR that is experienced by multiple identical users having the same connection characteristics, with respect to the bandwidth of the bottleneck link.

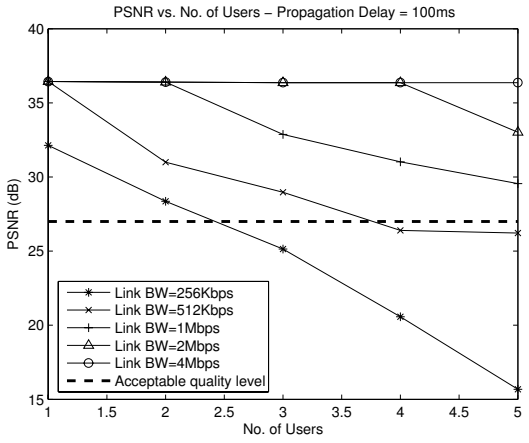


Fig. 8. Mean PSNR vs. Number of active users.

The dashed line illustrates the limit for acceptable video quality (27dB) as mentioned in Section 4.2. As the link bandwidth is high enough to sustain the aggregated video transmission rate, all users are supported by the video streaming server at equal quality levels. In particular, results show that at most two, three and four users can be supported at high quality (above 27dB) when the bottleneck link bandwidth is 256Kbps, 512Kbps, and 1Mbps respectively. If there is additional traffic, the number of the supported users will be intuitively smaller.

5 Conclusions and Future Work

In this paper we present a fuzzy-based adaptive video transmission approach specifically designed for scalable video streaming over the Internet. Our main objective is to combine NATs with CATs in order to achieve acceptable QoS levels in unpredictable mobile and wireless network environments. Thus, we introduce two new components: a feedback mechanism and a decision algorithm, that deal with layered video streams.

We evaluated our approach under various cross traffic patterns and our results indicate that the algorithm can finely adapt the video stream bit rate to the available bandwidth. Simulations showed that the proposed algorithm maintains responsiveness to various traffic patterns like CBR, FTP, and web-like cross traffic. In addition, we studied the effect of the link bandwidth and propagation delay on the QoS, and we discovered that the objective quality remains acceptable even in the presence of FTP and Web cross traffic. We demonstrated that our system is able to scale up offering graceful performance degradation and the same time the available bandwidth is fairly shared between active users who receive almost the same quality in terms of PSNR. We investigated the capacity of the system with respect to the number of users that can be supported by a video server. We showed that 2, 3 and 4 users can be supported at high quality when the bottleneck link bandwidth is 256Kbps, 512Kbps, and 1Mbps respectively.

For future work we are planning to provide a comparative study between our approach and other existing approaches in order to assess its advantages, by looking at the interaction between our adaptive flow and other flows sharing the same routers. In addition, we will investigate the capability of our approach to cope with handoff issues.

References

1. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003.
2. S. Floyd, "TCP and explicit congestion notification," ACM Computer Communication Review, Vol. 24, No. 5, October 1994, pp. 8-23.
3. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. on Networking, Vol. 1, August 1993.
4. C. Chrysostomou, A. Pitsillides, G. Hadjipollas, A. Sekercioglou and M. Polykarpou, "Fuzzy Explicit Marking for Congestion Control in Differentiated Services Networks," 8th IEEE Symposium on Computers and Communications (ISCC'03), 2003, pp. 312-319.
5. P. Antoniou, A. Pitsillides, and V. Vassiliou, Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Rate Control, 12th IEEE Symposium on Computers and Communications (ISCC'07), Aveiro, Portugal, July 1-4, 2007.
6. FFmpeg Multimedia System site, http://ffmpeg.mplayerhq.hu/.
7. Network Simulator (NS2), http://www.isi.edu/nsnam/ns/.
8. J. Klaue, B. Rathke and A. Wolish, "EvalVid - A Framework for Video Transmission and Quality Evaluation," Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Illinois, USA, September 2003, pp. 255-272.
9. C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, "An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission," Journal of Inf. Sc. and Eng. (accepted).
10. V. Vassiliou, P. Antoniou, I. Giannakou, and A. Pitsillides "Requirements for the Transmission of Streaming Video in Mobile Wireless Networks," International Conference on Artificial Neural Networks (ICANN), Athens, Greece, September 10-14, 2006.

An Experimental Analysis of the Mobile IPv6 Handover Latency Components

Vasos Vasilliou and Zinon Zinonos

Department of Computer Science, University of Cyprus,
Nicosia, Cyprus
{vasosv, zinonos}@cs.ucy.ac.cy

Abstract. This paper examines the handover process of Mobile IPv6 in a real wireless testbed, based on IEEE 802.11b and extracts information on the actions taken by network entities during the movement of a mobile node. This work focuses on the decomposition and analysis of all the initiated events and exchanged signals and measures, in a real-life scenario, the delays associated with them. Particular attention is given to the period leading to the L3 registration part of the handover, since this has been identified by many as the "choking point" of the whole process. Experimental results help in understanding the effect of Duplicate Address Detection, Router Advertisement Intervals and Wireless Beacon Intervals on the handover delay.

Keywords: Mobile IPv6, Handovers, L2 Delays, Router Advertisements, Beacon Intervals.

1 Introduction

Mobility is seen as an integral part of future networks, where the initiatives for next generation networks (NGN) will meet more traditional and established networks to form heterogeneous architectures. IPv6 is the networking technology of choice in the effort to move to an all-IP 4G environment. In order to effectively support the integration of Cellular (3G), Wireless LAN and Wireless Broadband (WiMax) technologies to the core networks, IPv6 and Mobile IPv6 will be required to provide transport and mobility solutions over different access technologies.

Mobility in IPv6 is therefore an enabler for future services and as such, all actions associated with it need to be thoroughly understood. One such action is the handover process. A handover (HO) is the process during which a mobile node (MN) creates a new connection and disassociates from its old one. The decision for a new association may be initiated due to movement, if we are moving away from the old connection point and we are approaching a new one; low signal quality, because of interference or other impairments in the wireless path; quality of service decision, trying to effect a balanced load among neighboring or overlapping cells; better service, if we recognize a network with services that we require; or policy and cost

decision, where the network or the user decide that it is more appropriate, or advantageous to relate to a different location.

Handovers can be characterized as Horizontal if they are performed between connection points using the same access technology, or Vertical if they are performed between access points of different technologies, a case which will be more common in future heterogeneous networks. In addition Handovers are considered Link Layer (L2) if they are performed between connection points belonging to the same subnet, or Network Layer (L3) if they are performed between different subnets and require the configuration of a different IPv6 address.

In this paper we examine the handover process of Mobile IPv6 in a real wireless testbed, based on IEEE 802.11b, and extract information on the actions taken by network entities during the movement of a mobile node. This work focuses on the decomposition and analysis of all the initiated events and exchanged signals and measures, in a real-life scenario all the delays associated with them, both in the Link and the Network layer.

Our work is important because it provides real-implementation results for significant parts of the handover process which cannot be obtained through simulation. We consider that simulators, though useful to some extent from an analytical point of view, either introduce unnecessary uncertainty into the network, or strictly specify significant parameters. Therefore there is always a margin of error in simulation results.

The rest of the paper is structured as follows: In Section 2 the mobility management, address resolution and other protocols related to MIPv6 handovers are presented. The handover process is analyzed in Section 3 and related work is outlined. The experimental evaluation of MIPv6 handovers in IEE802.11b is described in Section 4. Concluding remarks and items for future work are given in Section 5.

2 Protocol Overview

2.1 Mobile IPv6

Mobile IP version 6 is the mobile extension to IP version 6 [1]. The MIP fundamental principle is that a mobile node should use two IP addresses, a permanent address (the home address, assigned to the host and acting as its global identifier) and a temporary address (the care-of address -CoA, providing the host's actual location).

MIPv6 retains the general ideas of home network, encapsulation, home agent, and care-of address from MIPv4 [2]. However, it has a slightly different philosophy and a much-improved design than its predecessor.

While a mobile node is attached to its home network, it is able to receive packets destined to its home address, and being forwarded by means of conventional IP routing mechanisms. When the mobile node moves into a new network (visited/foreign network) its movement is detected and a new association is made with mobility agents (foreign agents) in the new domain. In MIPv6 a mobile node is more "independent" and does not have to rely on an access point to obtain a CoA and register with the home agent. To obtain its CoA, a mobile node uses the IPv6

protocols for Address Autoconfiguration [3] and Neighbor Discovery [4]. Figure 1shows the signaling taking place during MIPv6 handovers.

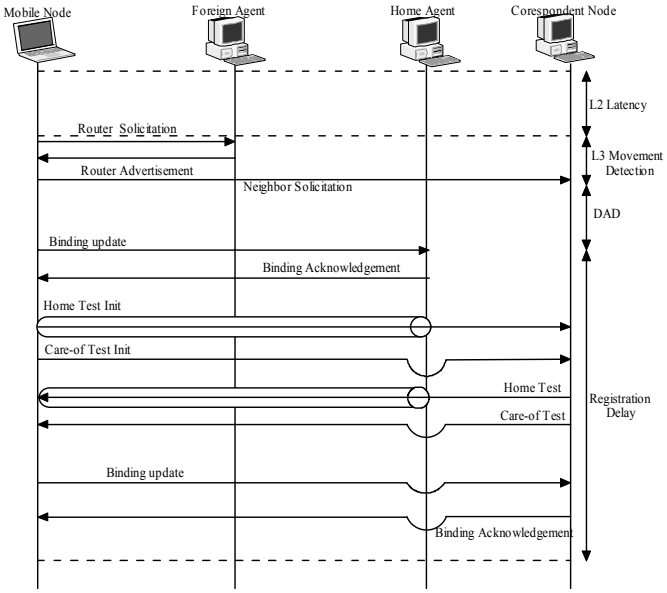


Figure 1 MIPv6 Handoff Signaling

Once configured with a CoA, the MN needs to send a Binding Update (BU) message to its Home Agent (HA) to register its current primary CoA. The first time a correspondent node (CN) needs to communicate with the MN, it sends the packets to the mobile node's home address. The HA is then encapsulating the packets and forwards them to the MN's CoA, where they are de-capsulated by the corresponding mobility agent and forwarded to the mobile node.

Upon a new association, the MN transmits BUs to its HA and the communicating CNs for them to associate the MN's home address with the new CoA. From then on, CNs will use IPv6 Routing headers for forwarding the packets to the MN. These packets have as destination address the MN's CoA. The 'home address' information is also included in the routing header to preserve transparency to upper layers and ensure session continuity. In the reverse direction, datagrams sent by the mobile node can be delivered to their destination using standard IP routing, without having to go through the home agent. Packets have as a source address the host's CoA while the home address is also included for the same reasons as above.

When the home agent discovers that the mobile node has moved, it uses techniques from Neighbor Discovery to indicate the new MAC address for the mobile node to all the correspondent nodes on the mobile node's home network.

Two well-known approaches in reducing the MIP handoff latency have been proposed in the literature. One aims to reduce the (home) network registration time through a hierarchical management structure, while the other tries to minimize the lengthy address resolution delay by address pre-configuration through what is known as the fast-handoff mechanism.

Hierarchical mobility management protocols, like Hierarchical MIPv6 (HMIPv6) [5] decide *when* to perform an action (registration in this case), whereas fast handover protocols, like Fast MIPv6 (FMIPv6) [6], address the problem of *how* to perform L3 actions in a faster way.

2.2 Stateless Address Autoconfiguration and Duplicate Address Detection

The stateless address autoconfiguration mechanism [3] allows a host to generate its own addresses in the following way. Access routers advertise prefixes that identify the subnet(s) associated with a link, while hosts generate an 'interface identifier' that uniquely identifies an interface on each subnet. A global address is formed by combining the two. The formation of an address must be followed by the Duplicate Address Detection (DAD) procedure in order to avoid address duplication on links where stateless address autoconfiguration is used. The address autoconfiguration is composed of the following steps:

The host generates a link-local address for its interface on a link. When in handoff, the host can use the same interface identifier as the one used in the previous link. It then performs DAD to verify the uniqueness of this address, i.e. the interface identifier on the new link. It uses the prefix(es) advertised by routers for forming a global address so as to be able to communicate with hosts other than the neighboring ones. During DAD, the host transmits a Neighbor Solicitation for the tentative linklocal address and waits for some specified delay (RetransTimer) [4] till it considers the address unique. DAD only fails if in the mean time, the host receives a Neighbor Advertisement for the same address, meaning that another host is using the being questioned address or if another host is in the progress of performing DAD for the same address and has also transmitted a Neighbor Solicitation.

2.3 Layer 2 handover

The IEEE 802.11 handover procedure is composed of three distinct phases: scanning, authentication, and reassociation. During the IEEE 802.11 handoff procedure the MN performs a channel scanning to find the potential APs to associate with. In the passive scan mode, each MN listens for beacon messages which are periodically sent by APs. In addition to the passive scan, each MN may broadcast a probe frame on the channel and receive probe responses from APs in the active scan mode. Regardless of scanning modes, all possible channels defined by the IEEE 802.11 standard (11 or 13 channels) are examined during a scan.

The scanning results in a list of APs that have been detected and it includes the related information for each detected AP, such as ESSID, the AP's MAC address, and the measured signal strength of each AP. Based on the scan result, the MN chooses an AP to associate with (usually the one with the highest signal strength). After that, the MN initiates the authentication procedure by transmitting the frames related to it. If the authentication phase is successfu1, the MN tries to re-associate with the AP by sending a reassociation request message to the AP. Then, the AP responds with a re-association reply message which contains the results of the reassociation. If

everything is successful, this phase becomes the last step of the handover. The length of the scanning procedure may vary from one implementation to the other but is generally considered to be the heaviest part of a Wireless LAN handover [7][8].

3 Handover Latency Analysis

A mobile node is unable to receive IP packets on its new association point until the handover process finishes. The period between the transmission (or reception) of its last IP packet through the old connection and the first packet through the new connection is the handover latency. The handover latency is affected by several components:

- ***Link Layer Establishment Delay (D_{L2}):*** The time required by the physical interface to establish a new association. This is the L2 handover between access routers.
- ***Movement Detection (D_{RD}):*** The time required for the mobile node to receive beacons from the new access router, after disconnecting from the old AR.
- ***Duplicate Address Detection (D_{DAD}):*** The time required to recognize the uniqueness of an IPv6 address.
- ***BU/Registration Delay (D_{REG}):*** The time elapsed between the sending of the BU from the MN to the HA and the arrival/transmission of the first packet through the new access router.

The overall handover process as well as the component delays identified above are presented in Figure 1. The handover delay for MIPv6 can analytically be computed as:

$$D_{MIPv6} = D_{L2} + D_{RD} + D_{DAD} + D_{REG} \quad (1)$$

Due to the differences in access networks, hardware, implementation versions and traffic, there can be no single value for the overall MIPv6 delay. Related values found in the literature vary from 1.3 sec in [9][10] to 1.9 sec in [11] [13], and 2.6 sec in [12]. It should be noted that only the last three refer to real implementations.

As it can been seen from Figure 1 and equation (1), the overall MIPv6 handover latency can be reduced by direct manipulation of a number of parameters. Solutions like HMIPv6 and FMIPv6 manage to reduce the BU/Registration Delay. In our work we focus on the other three delay components: the D_{L2}, D_{RD}, and D_{DAD}.

3.1 L2 delays

The values measured or considered in the literature for the D_{L2} delay are between 50ms [9] and a few hundred milliseconds [6]. In [14] and [15] the value is at 100ms. In [16] the range is from 100-300ms. In [11] the range is from 50-400ms. L2 delays are however very dependent on the physical medium and always exhibit great

variations. Since the scanning, or probing, delay is the most prevalent one during an L2 handover, we believe that it merits special attention. In this work we shorten the wireless beacon interval to values below 100ms in an effort to reduce D_{L2}.

3.2 Router Advertisements

Router Solicitations (RSol) and Router Advertisements (RA) help the MN identify that it has changed subnets and provide it with the necessary information for the creation of the new CoA. While in traditional IPv6, the values for RAs were in the order of 3 to 5seconds, for Mobile IPv6 these values need to be significantly lower. In this work, we change the RA interval in an effort to deduce the effect of it on D_{RD}.

3.3 Duplicate Address Detection

Once the MN discovers a new router and creates a new CoA it tries to find out if the particular address is unique. This process is called Duplicate Address Detection and it is a significant part of the whole IPv6 process, with very little room for improvement. In this work we evaluate MIPv6 HOs with this feature either enabled or disabled.

3.4 Related Work

Reducing the L2 probe delay is not a protocol issue, but an implementation issue. In [12] the authors examine different IEEE802.11-based network cards and propose the reduction of the MaxChannelTime to 100ms in order to reduce the effect of the probing procedure.

In [9] they recognize that the DAD time is significant during a handoff and they propose a scheme for HMIP in order to reduce the DAD time on handoff delay. The scheme is called Stealth-time HMIP (SHMIP) and assigns a unique on-link care-of address (LCoA) to each mobile node and switches between one-layer IPv6 and two-layer IPv6 addressing. In this mechanism when a mobile node sends a local BU, it also sends Bus to its home agent and correspondent nodes at the same time, using LCoA instead of RCoA. To further reduce packet losses, they also adopt pre-handoff notification to request previous mobility anchor points (MAP) to buffer packets for the mobile node.

In [17] they work specifically on the registration delay component. They make the assumption that the link layer delay can he considered equal to zero for link layer technologies supporting soft handover. They also consider the movement detection delay depends upon the frequency of router advertisement and could be large in a bandwidth constraint environment.

In [18] the total handover latency MIPv6 is found to be 5 seconds. Based on the author's assumptions, if the L2 handover takes about 1 second, then the remaining 4 seconds are used for the L3 handover. This happened because the minimum period of *RA* (Router Advertisement) was 3 seconds and the maximum period was 5 seconds

which corresponds to the default setting in *wired* IPv6. In MIPv6 these values are expected by the RFC to be smaller.

In [19] the authors use analytical models to evaluate MIPv4, MIPv6, FMIPv6, and HMIPv6 and compare their performances in terms of handover delay for VoIP services. They propose an adaptive timer for the retransmission of router solicitations, binding updates and other control signals, to replace the backoff timer usually found in MIP implementations. The results obtained using the adaptive timer technique show a 50% improvement compared to the fixed-timers option. However, these results are purely analytical and make specific static assumptions on the values of the different L2 and L3 component delays.

In [10] they do similar comparisons, utilizing a simulator instead of mathematical analysis. They compare Standard MIP, HMIP, FMIP, FHMIP and FFHMIP focusing on L3 HO values, and ignoring L2 and DAD delays.

The authors in [20] claim that none of the Fast or "assisted" methods of handover can be applied in IEEE 802.11 systems since such systems are based on the fact that the APs involved in a MN's reassociation can "anticipate" the handover before it is actually performed. However the 802.11 APs become aware of a MN's movement only after real occurrence of a reassociation event at the new AP. Other methods of shortening the movement detection delay are: (a) the MN pre-caches the IP information needed to perform the IP movement detection, without depending on the MIP advertisements for this purpose and (b) the APs are either pre-configured with information useful to perform movement detection for a newly connected MN, or obtain this information via periodic announces or other similar methods (centralized caching of the necessary information in each subnet).

We believe that the results of our research work can be of benefit to the work of others trying to characterize mobility management protocols, since it can help them utilize real-life values in their simulations, emulations, or equations.

4 Experimental Evaluation

The work items identified in Sections 3.1, 3.2 and 3.3 are evaluated experimentally in this Section. The testbed setup is explained in Section 4.1 and the results of each evaluation are analyzed separately afterwards.

4.1 Testbed Setup

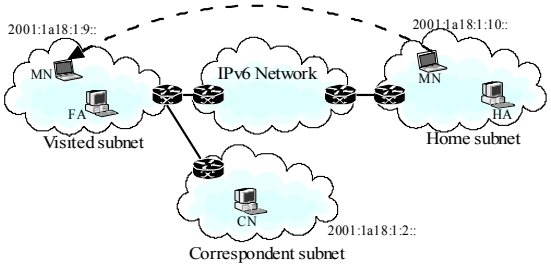


Figure 2 Testbed Topology

The experimental testbed consists of three wireless LANs connected through an IPv6 cloud as shown in Figure 2. This setup topology provides the simplest configuration for a realistic study of L2 and L3 handover components. Twenty experiments were run for each configuration and the average times are computed and analyzed. The devices used in the testbed have the specifications shown in Table 1.

Table 1 List of Equipment and Software

Mobile Node	Home Agent	Foreign Agent	Corresponded Node
IBM ThinkPad T42p	Acer Veriton 9100	Dell Optiplex GX1	Dell Optiplex GX1
Intel Pentium M 1.86GHz	Intel Pentium 4 1500MHZ	Intel Pentium III 50OMHz	Intel Pentium III 500MHz
2048 cache	256 cache	512 cache	512 cache
Atheros AR5212 802.11abg NIC	D-Link, PCI IEEE802.11b card, GWL-520, Atheros chipset		
Auto channel	Channel 1	Channel 6	
LINUX, Fedora Core 5, kernel 2.6.16			
MIPL v2.02			

Table 2 contains some of the configuration parameters in our testbed: Autoconfiguration is enabled, Forwarding is enabled only on the IPv6 routers and not on the MN, the MTU is 1500 bytes and the backoff timers for router solicitations, BU and Home / Co Test Initialization are set to the default values. The parameters on the right hand side are those changed in our experimental evaluation. The values in brackets are the default values.

Table 2 MIPv6 Testbed Parameters

Parameter	Value
mtu	1500
autoconf	1
forwarding	1 (MN=0)
Home / Co Test Init	1
Rt. Solicitation	1
BU	1.5

Parameter	Value
MinRouterAdv	0.03 - 1s (0.5)
MaxRouterAdv	0.07 - 1.5s (1.5)
DAD	On / Off (On)
Beacon Interval	50-100 ms (100)

4.2 Results

Based on the default values of Table 2, the mean total MIPv6 handover latency recorded in our setup was D_{MIPv6} = 3.68 sec. This delay is broken down as follows: D_{L2}+D_{RD}[1]=0.612s, D_{DAD} = 1.414s and D_{REG}= 1.651s

The major share in the handover latency goes to D_{REG} as expected. The BU and registration functions account for 45% of the total delay. The DAD function takes another 38% and the movement detection (including the L2 delay) accounts for the rest 17%. Compared to values recorded or computed by other researchers, the

[1] Due to the setup configuration, it was not possible to obtain separate values for D_{RD} and D_{L2}.

outcome of our measurements is higher by about one second. We believe that this is caused by slightly higher delays in all components, but especially in D_{REG} which contains the most transitions through the IPv6 cloud. It is important to mention that, to the best of our knowledge, no other work has taken this feature into account. Similar evaluations are made with the visited network directly connected to the home network and sometimes the visited/foreign router directly connected to a different interface of the HA, but none with a cloud of IPv6 nodes in-between.

4.2.1 Duplicate Address Detection

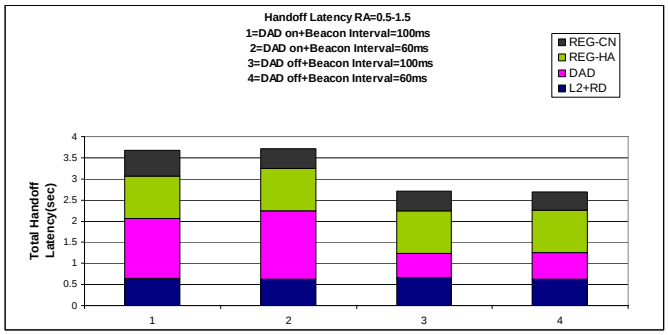


Figure 3 DAD component contribution to the MIPv6 Handover Latency

In Figure 3 we observe that when the DAD function is switched off the respective delay is reduced by almost 1sec which is the default timer value for this operation. In reality D_{DAD} is reduced on average by 0.835 and 0.979 sec for L2 wireless beacon intervals of 100ms and 60ms respectively.

Based on these results we can safely conclude that if we operate in a controlled environment where the probability of duplicate addresses is negligible, then we can discard the DAD function and achieve a decrease in the total MIPv6 delay of at most one second.

4.2.2 Router Advertisement Interval and Beacon Interval

The discovery of a new router is affected by two factors: the probe/scanning delay on L2 and the router discovery on L3. In this section we will examine the effect of the latter on the overall and component latencies in MIPv6.

Typical values for the min and max router advertisements are of the order of a few seconds in wired IPv6 networks. In MIPv6 these values are usually lowered and are centered around 1sec. We have started with our default values of 0.5 – 1.5 sec and lowered the intervals down to 0.03-0.07 sec.

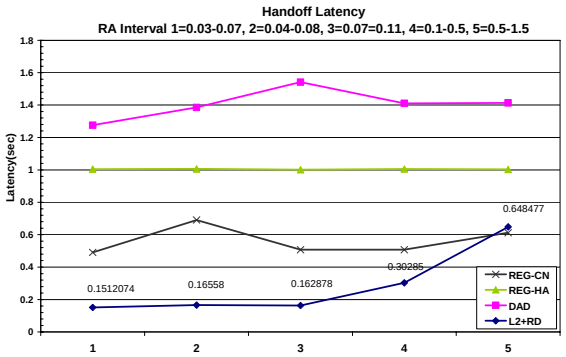


Figure 4. Router Advertisement Interval effect on handover component latencies.

In Figure 4 we recognize that the change in the RA interval only affects the combined D_{L2} + D_{RD} delay. We observe a 200-400% reduction in the corresponding delay between the default and lower values. This dramatic reduction is significant in terms of the handover delay, but may have other repercussions in the network which are not visible in these results. In this work we cannot comment yet on the effect of a lower interval on the overall network traffic and on the processing load of the routers.

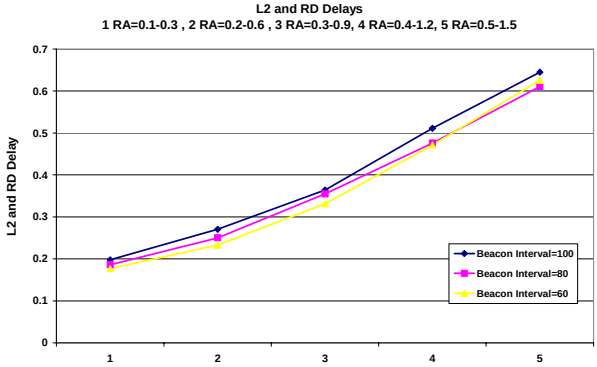


Figure 5. Router Advertisement and Beacon Interval

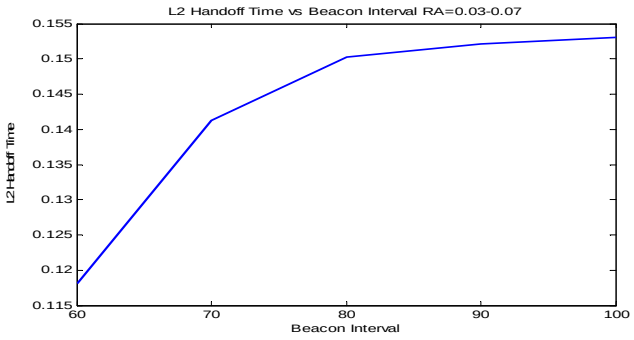


Figure 6. L2 Handoff Delay vs Beacon Interval

Based on the standards, MaxRouterAdv needs to be at least three times larger than the MinRouterAdv interval. In Figure 5 we adjust the ranges using MinRouterAdvintervals between 0.1 and 0.5sec. The result is again a 300% reduction in an almost linear manner. The figure is appended with plots of different Beacon Intervals, which do not provide any insight to their importance.

The Wireless Beacon Interval does make a change in the delay of the lower layer. Based on Figure 6 we see that a reduction of the Beacon Interval from 100ms to 60ms corresponds to an almost equal reduction in the L2 delay, with the steepest drop at 70 ms.

5 Conclusions and Future Work

In this paper we have examined the handover process of Mobile IPv6 in a real wireless testbed, based on IEEE 802.11b. This work has performed a detailed decomposition and analysis of the handover delay, with a focus on the pre-registration phase.

Our work is important because it provides real-implementation results for significant parts of the handover process which cannot be obtained through simulation. The testbed setup is considered to reference a very realistic topology and all the results were obtained with no optimizations on the L3 part of the implementations used. Our results illustrate how the link layer detection, the movement detection, and the address autoconfiguration parts of the handover can be reduced.

In addition, the outcome of this work can be utilized in recognizing further items for future research. Our future work will include the evaluation of the effect of the identified changes to the overall performance of the network. It also remains to be seen if the same delay reductions are present when the number of users in the network increases.

Acknowledgments. This work is partially funded by project NET6 (EPYΔΙ/0205/12) of the Cyprus Research Promotion Foundation, and by the FP6 IST Project MOTIVE. The authors would also like to thank Mr. Iacovos Pitharas for his help during the wireless testbed setup.

References

1. C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6, Request for Comment 3775, IETF, Jun. 2004.
2. C. Perkins Ed., “IP Mobility Support for IPv4”, Request for Comment 3344, IETF, Aug. 2002.
3. S. Thomson, T. Narten, T. Jinmei, and H. Soliman: IPv6 Stateless Address Autoconfiguration. Internet Draft, draft-ietf-ipv6-rfc2462bis-00.txt, February 2004.
4. T. Narten, E. Nordmark, and W. Simpson,: Neighbor Discovery for IP Version 6 (IPv6). Request for Comment 2461, IETF, December 1998.

5. H. Soliman, C. Castelluccia, K. El-Makri, and L. Bellier, ”Hierarchical Mobile IPv6 Mobility Management (HMIPv6),” Request for Comment 4140, Internet Engineering Task Force, Aug. 2005.
6. R. Koodli, et al. : Fast Handovers for Mobile IPv6, Request for Comment 4068, Internet Engineering Task Force, Jul 2005.
7. W-J Kim, D-H Kwon and Y-J Suh: Integrated Handoff Scheme of FMIPv6 and MAC Protocol in 4th Generation Systems. IEEE GLOBECOM 2005, vol.6, 28 Nov.-2 Dec. 2005
8. E. Ivov and T. Noel: An Experimental Performance Evaluation of the IETF FMIPv6 Protocol over IEEE 802.11 WLANs. IEEE WCNC 2006, April 2006
9. Wei Kuang Lai and Jung Chia Chiu: Improving Handoff Performance in Wireless Overlay Networks by Switching Between Two-Layer IPv6 and One-Layer IPv6 Addressing. IEEE Journal on Selected Areas in Communications, vol.23, no.11, November 2005
10. Y. Gwon, J. Kempf, and A. Yegin: Scalability and Robustness Analysis of Mobile IPv6, Fast Mobile IPv6, Hierarchical Mobile IPv6, and Hybrid IPv6 Mobility Protocols Using a Large-scale Simulation. IEEE International Conference on Communications 2004, vol.7, pp. 4087-4091, 20-24 June 2004
11. A.Mishra, et al.: An Empirical Analysis of the IEEE802.11 MAC Layer Handoff Process. Proc. of ACM SIGCOMM Computer Communication Review, Vol.33, Issue 2, Apr. 2003.
12. W. Lai, A. Sekercioglu, A. Pitsillides: Performance Evaluation of Mobile IPv6 Handover Extensions in an IEEE 802.11b Wireless Network Environment. 11th IEEE Symposium on Computers and Communications , Sardinia, Italy, June 26 –29, 2006, pp. 187-193.
13. A. Cabellos-Aparicio, H. Julian-Bertomeu, J. Núñez-Martínez, L. Jakab, R. Serral-Gracià, J. Domingo-Pascual: Measurement-Based Comparison of IPv4/IPv6 Mobility Protocols on a WLAN Scenario. Proceedings of the 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks. IEE UK, 2005.
14. S. Ryu1, Y. Lim, S. Ahn, and Y. Mun: Enhanced Fast Handover for Mobile IPv6 Based on IEEE 802.11 Network. ICCSA 2005, LNCS 3480, pp. 398 – 407, 2005.
15. S. Pack and Y. Choi : Performance Analysis of Fast Handover in Mobile IPv6 Networks. PWC 2003, LNCS 2775, pp. 679–691, 2003.
16. Y. Y. An, B. H. Yae, K. W. Lee, Y. Z Cho, and W. Y. Jung: Reduction of Handover Latency Using MIH Services in MIPv6. Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA’06).
17. I. Vivaldi, B.M. Ali, H. Habaebi, V.Prakash, and A. Salil: Routing Scheme for Macro Mobility Handover in Hierarchical Mobile IPv6 Network. 4th National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia
18. D-H Kwon, Y-S Kim, K-J Bae, and Y-J Suh: Access Router Information Protocol with FMIPv6 for Efficient Handovers and Their Implementations. IEEE GLOBECOM 2005
19. H. Fathi,, S. S. Chakraborty, and R. Prasad: Optimization of Mobile IPv6-Based Handovers to Support VoIP Services in Wireless Heterogeneous Networks. IEEE Transactions on Vehicular Technology, vol. 56, no.1, January 2007
20. I. Samprakou, C. J. Bouras and T. Karoubalis: Fast and Efficient IP Handover in IEEE 802.11 Wireless LANs. The 2004 International Conference on Wireless Networks, Las Vegas, Nevada, USA, 21 - 24 June 2004, pp. 249 - 255

Cross-layer performance modeling of wireless channels

D. Moltchanov

Institute of Communication Engineering,
Tampere University of Technology,
P.O.Box 553, Tampere, Finland
E-mail: moltchan@cs.tut.fi

Abstract. To optimize performance of applications running over wireless channels state-of-the-art wireless access technologies incorporate a number of advanced channel adaptation mechanisms at different layers of the protocol stack. These mechanisms affect performance provided to applications differently and their joint effect is often difficult to predict. Recently, to evaluate joint operation of various channel adaptation techniques, cross-layer performance models started to appear. These models abstract functionality of layers providing channel adaptation mechanisms and characterize performance of information transmission at data-link or higher layers, where it is usually standardized. In this paper we review recent cross-layer performance evaluation and optimization models, highlight their basic features and discuss applicability. Reviewed frameworks provide a starting point in cross-layer design of wireless channels.

1 Introduction

To optimize performance of applications in wired networks it is often sufficient to control performance degradation caused by packet forwarding procedures in network routers. Even though this is not a trivial task, dealing with wireless networks we also have to take into account performance degradation caused by incorrect reception of channel symbols at the air interface. These errors propagate to higher layers leading to performance degradation of applications. As a result, the air interface could be a 'weak point' in any quality of service (QoS) assurance model would ever be proposed for IP-based wireless networks. To deal with this problem state-of-the-art wireless access technologies incorporate a number of advanced features including multiple-in multiple-out (MIMO) antenna design, adaptive modulation and coding (AMC) scheme, automatic repeat request (ARQ) procedures, transport layer error concealment functionality, etc. To control and optimize functionality of various channel adaptation mechanisms a new cross-layer design paradigm is sought.

Up to date there were a number of proposals for cross-layers design of the protocol stack at the air interface. The common goal of all those proposals is to explicitly or implicitly exchange control information between different layers

whether at the runtime or at the design phase for further performance optimization purposes. However, most of those proposals were not supported by throughout cross-layer performance modeling studies of various organizations of the protocol stack. Indeed, the joint effect of various channel adaptation mechanisms is often difficult to model. Recently, such studies started to appear. It is expected that they may provide new theoretical and practical insights concerning what cross-layer interactions are necessary and sufficient for optimized operation of channel adaptation mechanisms.

The aim of this paper is to review cross-layer performance modeling and design frameworks proposed to date for various organizations of the protocol stack at the air interface. Although the literature on this topic exploded over the past few years, there are no approaches that are versatile enough to apply to any wireless technology. Instead, recent studies consider a certain combination of various adaptive mechanisms proposed for wireless channels. We highlight basic features, discuss applicability and review main conclusions of these frameworks.

The rest of the paper is organized as follows. In Section 2 we consider specifics of wireless channels. Cross-layer performance modeling frameworks are reviewed in Section 3. Conclusions are drawn in the last section.

2 Channel adaptation mechanisms

There are a number of factors that influence performance of wireless channels. The most important are traffic characteristics of applications, error-prone nature of wireless channels due to users movement and environmental characteristics of landscapes, and protocols with a set of their parameters, as shown in Fig. 1. Each application is characterized by its own traffic characteristics. Environmental characteristics of a landscape and movement of a user are stochastic factors determining propagation characteristics of a wireless channel. Protocols and their parameters determine how a given traffic is treated on a wireless channel. Evaluating performance that a given application experiences running over wireless channels is complicated task involving a number of interdependent stochastic and deterministic factors.

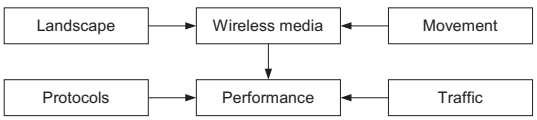


Fig. 1. Components that affect performance of wireless channels.

To provide stable performance to applications running over wireless channels, state-of-the-art wireless access technologies are expected to implement a number of advanced channel adaptation mechanisms. These are dynamic error concealment techniques, adaptive size of protocol data units at different layers, AMC

schemes, MIMO diversity, etc. At the application layer adaptive compression and coding (ACC) schemes can be enforced to reduce or increase the effective rate required from the network. These mechanisms are implemented at different layers of the protocol stack and affect performance provided to applications differently [1]. As a result, wireless access technologies call for novel design of the protocol stack that should now include cross-layer performance optimization and control functionality. Various mechanisms and their places in the protocol stack are marked with grey color in Fig. 2, where nRT stands for non-real-time applications, RT refers to real-time applications. To understand what performance level can be provided to applications, studies of joint operation of these mechanisms are required.

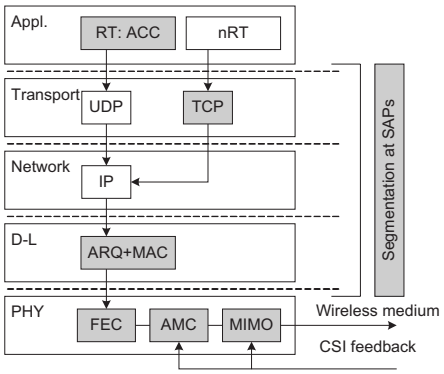


Fig. 2. Mechanisms affecting performance of wireless channels.

2.1 Multiple-in multiple-out system

Wireless channels are characterized by dynamic nature when the received signal strength varies in time, frequency and space. To effectively conceal channel fading problems that are frequent in wireless environment, time and frequency diversity techniques are already widely used. The spatial diversity is another type of diversity that currently receives a lot of attention from research community. This type of diversity is implemented by maintaining arrays of antennas at receiver and/or transmitters. Depending on the application area MIMO techniques are classified into two categories. First type of MIMO systems tries to improve throughput using spatial multiplexing. Second type is aimed at improving reliability of transmission. Nowadays, MIMO schemes are widely used in state-of-the-art wireless access technologies including UMTS, IEEE 802.16, etc.

There are a number of transmit/receive diversity schemes. When channel state information (CSI) in terms of the signal-to-noise (SNR) ratio is available at both sides of a link maximum ratio transmission (MRT) and maximum ratio

combining (MRC) are optimal transmit and receive diversity schemes. When no CSI is available at the transmitter, space-time block coding (STBC) can be used for transmit diversity. In this case selection combining (SC) can also be used for transmit and receive diversity.

2.2 Adaptive modulation and coding

When CSI is constantly fed back to the transmitter, AMC can be used to improve performance of information transmission. Generic AMC system works as follows. The range of the received SNR is divided into a number of intervals. These intervals are chosen such that a certain combination of constellation and coding rate provide the best possible spectral efficiency in a certain range of SNR. CSI in terms of the received SNR is constantly fed back to the transmitter on a frame-by-frame basis. When SNR changes, new constellation and coding rate are chosen and further used at the wireless channel. Note, that AMC itself provides significant gains in terms of optimal channel usage. Combined with MIMO, AMC may provide good results even in complicated propagation environments.

As one expects performance of AMC heavily depends on accuracy of SNR estimation at the receiver. Additionally, to effectively use AMC system the channel fading process must be slower than the SNR feedback sent from the receiver to transmitter. Nowadays, AMC system is used in many state-of-the art wireless access technologies including UMTS, IEEE 802.16, HIPERLAN/2, etc.

2.3 Error mitigation

Wireless channels are error-prone transmission media. Data-link layer error concealment techniques such as forward error correction (FEC) and automatic repeat requests (ARQ) are crucial for satisfactory performance of wireless channels. ARQ eliminates the influence of bit errors allowing to retransmit incorrectly received frames. To notify the sender about the erroneously received frame, ARQ protocols require a feedback channel. We distinguish between Stop-and-Wait (SW), Go-Back-N, and Selective Repeat (SR) ARQ schemes. According to the former approach the source transmits a frame and then waits for acknowledgement frame from the receiver. Go-Back-N ARQ is a scheme where frames are consecutively transmitted. When a certain frame is incorrectly received the receiver asks to retransmit all frames starting from incorrectly received one. According to SR-ARQ scheme only incorrectly received frames are requested. When the channel conditions are relatively 'bad' ARQ may introduce significant delays that are not always tolerable for delay-sensitive applications.

FEC procedures use proactive approach eliminating the influence of bit errors in advance, introducing error correction redundancy. This redundancy is exploited at the receiver to recover from bit errors. The major advantage of FEC schemes is that they do not introduce long delays allowing some information to be lost. Depending on a particular technology, FEC capabilities can be implemented at the physical or data-link layers. Due to complementary advantages, FEC and ARQ are often used in combination.

In spite of lower layers' error correction techniques, errors may still propagate to higher layers. In addition to data-link and physical layer error concealment, transport layer protocols, such as TCP, also perform error correction. Joint operation of error concealment techniques at the data-link (ARQ) and transport (TCP) layers was studied by Zorzi and Rao in [2, 3]. Among other conclusions, authors demonstrated that interworking between different layers performing error correction may significantly improve performance provided to applications.

2.4 Medium access control

In wireless networks we distinguish between centralized and distributed medium access control (MAC) schemes. The former scheme adopts a cental coordination entity that assign the channel to a particular mobile station for the whole duration of transmission. No such entity is available in distributed access environment. From the performance modeling point of view centralized MAC schemes can be treated similarly. Indeed, in all schemes a certain amount of transmission resources is exclusively assigned to a mobile station. ALOHA, slotted ALOHA and collision avoidance multiple access (CSMA) scheme are examples of distributed MAC. Among the wide family of CSMA protocols, CSMA with collision avoidance (CSMA/CA) is widely used in wireless local area networks (WLAN). Performance models are different for centralized and distributed access schemes.

3 Cross-Layer Performance Modeling

3.1 Layer of interest

Performance of wireless channels is usually estimated at the data-link layer. There are a number of reasons behind that. First of all, data-link layer incorporates medium access procedures that significantly affect performance provided to higher layers. Secondly, channel adaptation mechanisms and local error correction procedures are also defined for the data-link or physical layers. From this point of view, performance models at the data-link layer abstracts functionality of underlying protocols describing performance provided to higher layers.

3.2 Basic principles

The received signal strength, SNR, and bit error models cannot be directly used in performance evaluation studies at the data-link or higher layers and must be previously extended to the layer at which performance of applications is evaluated and standardized. For such an extension to be accurate, we have to take into account specific peculiarities of underlying layers including physical layer mechanisms such as MIMO and AMC, data-link error concealment techniques, segmentation procedures between adjacent layers, etc.

Basic principles of cross-layer wireless channel modeling are illustrated in Fig. 3, where black rectangles denote incorrectly received protocol data units

(PDU), grey rectangles stand for correctly received PDUs. Here, we assume that FEC is capable to conceal at most one incorrectly received bit and ARQ is not used at the data-link layer. Since no error correction procedures are defined for IP layer, even a single lost frame within a packet leads to loss of a whole packet. Following frameworks proposed in [4, 5] performance parameters at the IP layer can be obtained as the function of underlying layers parameters. More restrictive assumptions are required to incorporate ARQ, AMC and MIMO functionalities.

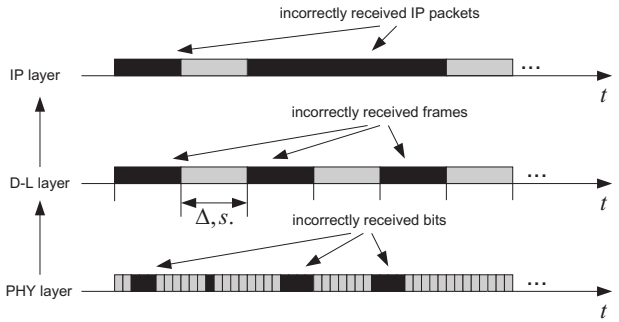


Fig. 3. Cross-layer performance modeling of wireless channels.

3.3 Frameworks for real-time applications

The work of Li *et. al* [4] was the first where authors tried to explicitly represent wireless channel performance at the IP layer as a cross-layer function of mechanisms implemented at the physical and data-link layers. They started with SNR observations and went up to data-link and IP layers defining frame error and packet error processes as a function of underlying layer's statistics and properties of protocols. The channel was assumed to operate according to centralized MAC scheme meaning that it is exclusively assigned to a certain user during the whole duration of information transmission. Unfortunately, the proposed approach is limited to mean values of error processes at different layers and neglects second-order properties of error statistics. As was previously pointed out by these authors, second-order statistical characteristics may have a pronounced impact on performance measures of the service process. Indeed, according to Li and Hwang [6] the major impact on performance parameters of the service process is produced by the empirical distribution of the arrival process and the structure of its autocorrelation function (ACF). Hayek and He [7] highlighted importance of empirical distributions of the number of arrivals showing that the queuing response may vary even for inputs with the same mean and ACF. For wireless channels second-order properties of the channel process have similar significance as those of the arrival process in error-free environment

[8]. Finally, the proposed approach requires too many unrealistic assumptions about organization of the wireless channels.

In [9] Lui *et. al* developed an analytical performance evaluation framework for a wireless channel with AMC at the physical layer and truncated ARQ at the data-link layer. The channel process was modeled by finite-state Markov chain (FSMC) where each state corresponds to a certain SNR range and the middle of a range is chosen as the current value of SNR process. The number of ranges was limited to $K = 7$ resulting in 8 boundary points denoted by $\{T_k, k = 0, 1, \ldots, K\}$ such that $T_0 < T_1 < \cdots < T_k$, $T_0 = 0$, $T_k = \infty$. Code rates in these modes are 0, 0.5, 0.5, 0.75, 0.5625, 0.75 and 0.75, respectively. Corresponding constellations are binary phase shift keying (BPSK), quaternary PSK (QPSK), 16-quadrature amplitude modulation (QAM), 16-QAM, and 64-QAM. With codes rates ranging from 0 to 0.75 and modulation ranging from BPSK to 64-QAM the spectral efficiency of each mode is 0, 0.5, 1.0, 1.5, 2.25, 3.0 and 4.5 b/s/Hz. When SNR gets better, a faster mode is selected. When SNR gets worse, the lower spectral efficiency is chosen. For any given frame error rate (FER) and performance requirements of a service, the boundaries $\{T_k, k = 0, 1, \ldots, K\}$ can be derived numerically using probability density function of the SNR as shown in [9]. When CSI is enabled and assumed to fed back timely, this model captures changes in modulation schemes and code rates. As a result, each state of the FSMC can be associated with a certain bit error probability. The channel was assumed to be flat fading and remains constant during frame transmission. As a result, AMC is adjusted on a frame-by-frame basis. Authors assumed buffering at the data-link layer. When a frame is received and the buffer is full it is lost. Due to presence of AMC the service process associated with transmit buffer is time-varying in nature with FSMC capturing its dynamics. The arrival process was assumed to be Poisson in nature that limits application area of the proposed methodology. In this case the queuing system of interest is described by two-dimensional Markov chain. When arrivals are assumed to be more complicated and follow, for example, doubly-stochastic Markov process the analysis would require three-dimensional Markov chain which is hardly to analyze numerically for steady-state probabilities. Performance parameters of interest were the frame loss rate and throughput of the system.

Provisioning of performance guarantees for applications running over wireless channels with AMC and ARQ functionalities was considered in [10]. The problem solved in [10] is dual to that one considered in [9]. Particularly, authors provided a way to estimate a minimum amount of bandwidth required to serve Poisson arrivals sent over a wireless channel with AMC and truncated ARQ procedures. Results of Lui *et. al* are summarized in [11, 12]. In [11] authors carried out numerical experiments revealing that the strength of spectral efficiency using ARQ at the data-link layer is compared to that provided by diversity given that the maximum number of transmissions per packet equals the diversity order. However, they also pointed out that diversity could be of special importance for real-time applications that usually pose a limit on the maximum number of retransmissions. Based on obtained results authors in [11, 12] provided insights

into the cross-layer system design that takes into account interactions between wireless channel and queuing process.

Scheduling for performance guarantees over wireless channels with AMC is considered in [13]. To the best of our knowledge [13] is the first study where optimization task was dealt with. In that contribution a multiuser scenario at the data link layer was considered, where each user employs AMC at the physical layer. Two service classes were assumed, namely, QoS-guaranteed and best-efforts with each user classified to a certain class. The cross-layer scheduler proposed by authors benefits from low-complexity implementation and analysis, provides service isolation and scalability, decouples delay from dynamically-scheduled bandwidth, and is backward compatible with existing separate-layer designs. The performance evaluation framework employed by authors is similar to that one proposed in [11, 12].

Very important contributions came from Zhang *et. al* [14, 8]. To the authors' knowledge those were first studies where MIMO functionality of the physical layer was modeled using unified analytical framework. Particularly, in [14] it was demonstrated that for a variety of MIMO spatial diversity schemes the probability density function (pdf) can be obtained as an unified expression. In [8] authors reviewed basic principles of the cross-layer modeling methodology and proposed modeling framework to investigate the impact of physical layer parameters on performance delivered to applications at the data-link layer. In this contribution both MIMO and AMC were assumed at the physical layer. The MIMO system was assumed to provide spatial diversity. Using results of [14] the service process at the physical layer with MIMO and AMC enabled was modeled by FSMC similar to [9]. Authors assumed that the channel state remains the same during the frame transmission time but may vary from frame to frame. Performance parameters of interest were limited to delay bound and buffer overflow probability and evaluated using so-called effective capacity approach.

The effective capacity is the dual problem to well-known effective bandwidth. The latter has been extensively studied in the beginning of 90s. Recall, that for any stationary arrival process, relatively large buffer space, K, and constant service rate, the overflow probability is given by $Pr\{Q > K\} \approx e^{-\theta K}$, where θ is called QoS exponent [8]. When K is small the following approximation is used $Pr\{Q > K\} \approx \alpha e^{-\theta K}$, where α is the probability that the buffer is not empty. The delay bound is then given by $Pr\{D > \tau_{\max}\} \approx \epsilon e^{-\theta \delta \tau_{\max}}$, where $\tau_{\max}$ is the delay bound. Note, that the service process of wireless channels is variable in nature. Wu and Negi in [15–17] formulated and solved the dual problem of finding a maximum arrival traffic that can be supported over a given wireless channels such that performance parameters of interest (e.g. delay bound) are satisfied. The effective capacity and effective bandwidth allow to analyze the statistical delay-bound violation probability, which is of paramount importance for wireless networks. However, the concept of effective capacity as was proposed in [15–17] assumes a constant arrival traffic, which is not realistic for most practical applications. In [8] authors extended the model allowing arbitrary stationary service and arrival processes. Particularly, it was shown that the effective capacity of a

wireless channel modeled by FSMC is given by [8]

$$E(\theta) = -\frac{1}{\theta}\log\Big(\rho\big\{P\Phi(\theta)\big\}\Big), \tag{1}$$

where θ is the QoS exponent, P is the one-step transition probability matrix governing FSMC, $\rho\{\cdot\}$ is the spectral radius of P and $\Phi(\theta)$ is given by

$$\Phi(\theta) = diag(e^{-\mu_1\theta}, e^{-\mu_2\theta}, \ldots, e^{-\mu_K\theta}), \tag{2}$$

where μ_i, $i = 1, 2, \ldots, K$ is the number of bits transmitted in the state i of the FSMC, K is the number of states.

Cross-layer performance modeling methodology was also proposed in [18], where authors investigated influence of physical layer parameters on performance of higher layers. Performance parameters of interest were channel holding time, handoff probability, handoff call arrival rate, call blocking probability, call completion probability, and forced termination probability. The channel parameters were limited to the carrier frequency, maximum Doppler frequency, and fading margin. Among other conclusions, authors demonstrated that fading channel characteristics affect considered teletraffic variables.

In a number of contributions [19, 5, 20] Moltchanov *et. al* proposed a cross-layer performance modeling framework that takes into account FEC and ARQ procedures implemented at the data-link layer. The wireless channel was assumed to provide CBR service and modeled as a special case of FSMC. Performance parameters of interest were delay and loss probability distribution functions (pdf) at the IP layer. The major advantage of the proposed framework is that it explicitly takes into account memory of both wireless and arrival processes. Specifically, the arrival process is allowed to be as arbitrary as discrete-time batch Markovian arrival process (D-BMAP), which were shown to be quite versatile in terms of modeling capabilities (see [21, 22] among others). Since the wireless channel model is allowed to be arbitrary FSMC, MIMO and AMC functionality of the physical layer can also be taken into account without significant increase of complexity of analysis. Among other conclusions, authors demonstrated that first and second-order statistical characteristics of arrival and service processes have a profound impact on performance metrics of interest, especially when these processes are autocorrelated in nature. Particularly, it was revealed that when channel process is only slightly autocorrelated the gain provided by FEC codes is negligible.

3.4 Frameworks for non-real-time applications

Performance of various TCP versions in presence of correlated errors at the data-link layer was studied by Zorzi *et. al* in [23, 24] using cross-layer approach. In those studies the system of interest incorporates ARQ functionality of the data-link layer and congestion avoidance and control features of TCP. Each TCP segment was assumed to be divided into a number of data-link layer frames. In [23] it was demonstrated that clustering (correlation) of errors at the data-link layer

positively affect throughput obtained by TCP Tahoe compared to completely random errors. Similar observations have been found in [24] where authors compared performance of various TCP versions. Particularly, they demonstrated that TCP Tahoe and TCP Reno perform similarly over slowly fading channels characterized by strong memory. TCP Tahoe and TCP NewReno also perform almost similar in that environment. These results are in accordance with later findings made for UDP protocol presented in [5, 20] and questions the need for memory removal techniques such as interleaving. It is important to note that in these contributions authors assumed Gilbert-Elliott model of the wireless channel at the data-link layer [25]. Although the channel model can be extended to FSMC capturing the presence of MIMO and AMC systems at the physical layer, this results in significant increase in complexity of analysis. Indeed, three-dimensional Markov chain is required to describe TCP performance.

Cross-layer performance modeling framework that explicitly takes into account MIMO design of transceivers was proposed in [26]. As a channel model authors assumed Gilbert model [27] and analyzed dependencies between both TCP sending rate and UDP performance and various wireless channel characteristics including the Doppler effect and optimal coding schemes. The proposed framework was used to analyze the TCP performance of two MIMO systems, namely, the BLAST system and the STBC system. To calculate Gilbert model parameters assuming MIMO system simulation approach has been taken. The TCP model was adopted from [28]. Authors demonstrated that while the optimal rate for maximum TCP throughput is far from the channel capacity, the optimal rate for error and delay-tolerant video transmission requires much higher rates. As a result, the physical layer should be able to adapt to the type of application in order to increase the system performance advocating the need for cross-layer design of the protocol stack. They also revealed that mobility benefits systems with larger buffers, especially for TCP, as the ARQ scheme is able to recover the shorter burst errors. In overall, authors showed that the applications that uses wireless transmission medium plays a critical role in choice of optimal lower layer parameters advocating the need for cross-layer design of the protocol stack at the air interface.

Liu *et. al* also considered performance of TCP transmission over wireless channels with AMC implemented at the physical layer, finite queue length and truncated ARQ at the data-link layer [29]. Assumptions regarding physical and data-link layers were similar to those taken in [9, 30]. The considered TCP Reno model was adopted from [28]. The only modification made to the PFTK model was that the delay bound at the wireless part was added to the final expression. Note that later the model proposed in [28] was shown to overestimate the actual TCP performance [31]. However, qualitatively results of [29] remain valid.

4 Conclusions

To provide best possible performance, state-of-the-art wireless access technologies incorporate a number of dynamic mechanisms that allow to change opera-

tional parameters of various protocols on-the-fly. However, modification to any component of the system does not only change its own behavior but may also affect the performance of the whole system. Results of this influence are often difficult to predict. Additional efforts are required to ensure stability of the system. In this paper we reviewed cross-layer layer performance modeling frameworks proposed to date for performance evaluation of wireless channels. These frameworks provide a starting point in cross-layer design of wireless channels describing joint performance of two or more channel adaptation schemes.

Summarizing, we note that performance of centralized MAC schemes was studied fairly well. There are models capturing joint behavior of MIMO, AMC and error correction mechanisms. Although all those frameworks proposed so far capture two or more error mitigation mechanisms the lack of performance optimization studies is evident. Additionally, performance of distributed MAC schemes implementing dynamic error mitigation mechanisms has not been addressed so far. It can be partially explained by comprehensive behavior of those channel access schemes.

References

1. V. Srivastava and M. Motani. Cross-layer design: a survey and the road ahead. *IEEE Comp. Comm.*, 43(12):112–119, Dec. 2005.
2. M. Zorzi and R. Rao. Error propagation in protocol stacks. In *In Proc. ISI'97*, page 262, Ulm, Genrmany, June/July 1997.
3. M. Zorzi and R. Rao. Error control in multi-layered stacks. In *In Proc. GLOBECOM*, pages 1413–1418, 1997.
4. Y.-Y. Kim and S.-Q. Li. Capturing important statistics of a fading/shadowing channel for network performance analysis. *IEEE JSAC*, 17(5):888–901, May 1999.
5. D. Moltchanov, Y. Koucheryavy, and J. Harju. Cross-layer modeling of wireless channels for IP layer performance evaluation of delay-sensitive applications. *Comp. Comm.*, 29(7):827–841, Apr. 2006.
6. S.-Q. Li and C.-L. Hwang. Queue response to input correlation functions: discrete spectral analysis. *IEEE Trans. Netw.*, 1:522–533, Oct. 1997.
7. B. Hajek and L. He. On variations of queue response for inputs with the same mean and autocorrelation function. *IEEE Trans. Netw.*, 6(5):588–598, Oct. 1998.
8. X. Zhang, J. Tang, H.-H. Chen, S. Ci, and M. Guizani. Cross-layer-based modeling for quality of service guarantees in mobile wireless networks. *IEEE Comm. Mag.*, pages 100–106, Jan. 2006.
9. Q. Lui, S. Zhou, and G. Ganniakis. Combining adpative modulation and coding with truncated ARQ enhances throughput. In *In IEEE Workshop on Signal Processing*, pages 110–114, 2003.
10. Q. Lui, S. Zhou, and G. Ganniakis. Efficient bandwidth utilization guaranteeing QoS over adaptive wireless links. In *In Proc. IEEE Globecom*, pages 2684–2688, 2004.
11. Q. Lui, S. Zhou, and G. Ganniakis. Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links. *IEEE Trans. Netw.*, 3(5):1746–1755, Sept. 2004.
12. Q. Lui, S. Zhou, and G. Ganniakis. Queuing with adaptive modulation and coding over wireless links: Cross-layer analysis and design. *IEEE Trans. Wir. Comm.*, 4(3):1142–1153, May 2005.

13. Q. Lui, S. Zhou, and G. Ganniakis. Cross-layer scheduling with prescribed QoS guarantees in adaptive wireless networks. *IEEE JSAC*, 23(5):1056–1066, May 2005.
14. J. Tang and X. Zhang. Capacity analysis for integrated multiuser and antenna diversity over nakagami-m fading channels in mobile wireless networks. In *In Proc. CISS'05*, 16-18 2005.
15. D. Wu and R. Negi. Effective capacity: a wireless link model for support of quality of service. *IEEE Trans. Veh. Tech.*, 2(4):630–643, July 2003.
16. D. Wu and R. Negi. Downlink scheduling in a cellular network for quality-of-service assurance. *IEEE Trans. Veh. Tech.*, 53(5):1547–1557, Sep. 2004.
17. D. Wu and R. Negi. Utilizing multiuser diversity for efficient support of quality of service over a fading channel. *IEEE Trans. Veh. Tech.*, 54(3):1198–1206, May 2005.
18. Y. Zhang and M. Fujise. Performance analysis of wireless networks over rayleigh fading channel. *IEEE Trans. Veh. Tech.*, 55(5):1621–1632, Sep. 2006.
19. D. Moltchanov, Y. Koucheryavy, and J. Harju. Cross-layer analytical modeling of wireless channels for accurate performance evaluation. In *QoFiS 2004*, pages 194–203, Barcelona, Spain, Sept.–Oct. 2004.
20. D. Moltchanov, Y. Koucheryavy, and J. Harju. Loss performance model for wireless channels with autocorrelated arrivals and losses. *Comp. Comm.*, 29(13–14):2646–2660, Aug. 2006.
21. J. Cosmas, G. Petit, R. Lehnert, C. Blondia, K. Kontovvassilis, O. Casals, and T. Theimer. A review of voice, data and video traffic models for ATM. *European Transactions on Telecommunication and Related Technologies*, 5(2):139–153, March–April 1994.
22. C. Blondia. A discrete-time batch Markovian arrival process as B-ISDN traffic model. *Belgian Journal of Oper. Res.*, 32(3,4):3–23, 1993.
23. M. Zorzi and R. Rao. The effect of correlated errors on the performance of TCP. *IEEE Comm. Let.*, 1(5):127–129, Sep. 1997.
24. M. Zorzi, A. Chockalingam, and R. Rao. Throughput analysis of TCP on channels with memory. *IEEE JSAC*, 18(7):1289–1300, July 1999.
25. E. Elliott. Estimates of error rates for codes on burst-noise channel. *Bell Systems Technical Journal*, pages 1977–1997, 1963.
26. A. Toledo, X. Wang, and B. Lu. A cross-layer TCP modelling framework for MIMO wireless systems. *IEEE Trans. Wir. Comm.*, 5(4):920–929, Apr. 2006.
27. E. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, 1960.
28. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP reno performance: a simple model and its empirical validation. *IEEE Trans. Netw.*, 8(2):133–145, Apr. 2000.
29. Q. Lui, S. Zhou, and G. Ganniakis. TCP performance in wireless access with adaptive modulation and coding. In *In Proc. ICC*, pages 3989–3993, 2004.
30. Q. Lui, S. Zhou, and G. Ganniakis. Cross-layer combining of queuing with adaptive modulation and conding over wireless links. pages 717–722.
31. R. Dunaytsev, Y. Koucheryavy, and J. Harju. The PFTK-model revised. *Comp. Comm.*, 29(13/14):2671–2679, Aug. 2006.

Indoor location for safety applications using wireless networks

F. Barceló-Arroyo[1], M. Ciurana[1], I. Watt[2], F. Evenou[2], L. De Nardis[3], P. Tome[4]

[1] Universitat Politècnica de Catalunya
[2] France Telecom R&D
[3] University of Rome La Sapienza
[4] Swiss Federal Institute of Technology

Abstract—This paper presents the indoor positioning research activities carried out within the scope of the Liaison project. Most of the work has been performed on WiFi location. WiFi is nowadays widely deployed in buildings such as hotels, hospitals, airports, train stations, public buildings, etc. Using this infrastructure to locate terminals connected to the wireless LAN is expected to have a low cost. Methods presented in this paper include fingerprinting with particle filter constrained on a Voronoi diagram and TOA based on data frames and acknowledgments at the IEEE 802.11 MAC level. Other technologies have also been researched: A-GNSS to handle the transition between outdoors and indoors, UWB in ad-hoc mode to cope with possible lacks of infrastructure and inertial MEMS to increase the availability and robustness of the overall system.

1 Introduction

1.1 Location Based Services

Recently, there has been a growing interest on location-based services (LBS). LBS are particularly relevant in the case of mobile networks. They can be defined as services that adapt to a user's location and situation: location is thus a crucial input for these applications. LBS explore the ability of technology to know where the user is and shape the information provided accordingly. Presently, many LBS have already been deployed and others that have been designed are ready for commercial implementation. A few of the most interesting ones are: information services, navigation, workforce management, demand-responsive transport, lone worker applications, children tracking, medical alert…

1.2 Indoor location

Outdoors is the typical scenario for GPS positioning and tracking. When the terminal to be located has an open view of the sky, GPS is expected to give good or even excellent accuracy. Difficulties with GPS positioning usually occur in urban canyons and indoors, where it is difficult or impossible to acquire the necessary satellites for a position computation. On the other hand, research has been performed on location systems that use

the cellular network (GSM-GPRS, UMTS…) to provide the terminal's position, but the main drawback is that they do no provide enough accurate positioning for many of the location based services. Given the situation, there is the need to research on alternative location techniques that are able to provide accurate location information of the user in medium to deep indoors, electrically noisy indoor scenarios, subterranean places (e.g. parking), etc, because in many of the location based services mentioned in the previous section the people to be located are in such environments. As WiFi is nowadays widely deployed in buildings such as hotels, hospitals, airports, train stations, public buildings, etc, it seems to be a suitable infrastructure to provide cost-efficient positioning solutions.

2 Fingerprinting in WLAN

2.1 Introduction

This technique aims at taking the major advantage of one of the available outputs of a standard WiFi card, which is the received signal strength (RSS) from each Access Point (AP). Given this consideration, it is possible to get a list of the received power coming from all the APs covering the area where the mobile is moving. The simplest approach for locating a mobile device in a WLAN environment using this available information is to approximate its position by the position of the APs received at that position with the strongest signal strength, but its main drawback is its large estimation error. The accuracy is inversely proportional to the range of APs which is within 25 and 50 meters for indoor environments [3]. On the other hand, using a propagation model [1][2] to turn RSS measurements into distances did not provide satisfying results when introducing these ranges in a multilateration algorithm. [4] introduces a different approach for locating the device in indoor environments by using the radio signal strength fingerprinting.

Fingerprinting mainly consists in having some signal power footprints or signatures that define a position in the environment. This signature consists of the received signal powers from different APs that cover the environment. In a first step, called training for profiling, it is necessary to build this mapping between collected received signal strength and certain positions in the environment. This leads to a database that is used during the positioning phase. Building the footprint database can be done in two ways. A first method is to do on-site measurements for some reference positions in the building with a user terminal. An alternative approach is based on collecting limited on-site measurements and introducing them in an adjustable propagation model that would use them to fit some of its parameters. Then, this propagation model gives an extensive coverage map for each AP. However, the poor results obtained earlier with the use of the propagation model did not invite us to focus on such approach. Neural networks are another learning method for improving propagation models over time [5]. Ray tracing tools represent another solution to build such a database, but they are very complex. Moreover, a good knowledge of the radio environment

(knowledge of the presence and position of all the APs) is needed to cope with the interfering issue. However, such information is not always available due to the fast growing emergence of this technology in indoor environments. It was decided then to carry on with the use of data collection to build the database.

Once this prerequisite step is accomplished, it is necessary to perform the reverse operation, which will deliver the position associated to an instantaneous collected tuple of received signal strengths. Different techniques can fit these requirements. One of the simplest ones is the k-closest neighbors algorithm, which goes through the database and picks the *k* referenced positions that match best the observed received signal strength tuple. The criterion that is commonly retained is the Euclidian distance (in signal space) metric. The estimated position of the mobile is considered to be the barycentre of those *k* selected positions. The main advantage of this method is its simplicity to set it up. However, the accuracy highly depends on the granularity of the reference database [6]. A better accuracy can be achieved with finer grids, but a finer grid means a larger database which is more timecostly. However, in both techniques, the signal strength fluctuations introduce many unexpected jumps in the final trajectory. Removing those jumps can be done by using a filter. Kalman filter and particle filter are often used in parameter estimating problems and tracking. This last filter will be introduced in the next section, and the benefits for using such a filter will be presented.

2.2 Improving WiFi positioning with a particle filter constrained on a Voronoi diagram

Nowadays, maps of most public or corporate buildings are available in digital format (dxf, jpeg, etc). The key idea is to combine the motion model of a person and the map information in a filter, in order to obtain a more realistic trajectory and a smaller error for a trip around the building. In the following, it will be considered that the map available is in bitmap format. So, no other information is available except for the pixels in black and white which depict the structure of the building. The particle filter, based on a set of random weighted samples (i.e. the particles), represents the density function of the mobile-position. Each particle explores the environment according to the motion model and map information. Their weights are updated each time a new measurement is received. However, the free particle filter is not fit for handset based applications, as the computations are quite heavy. At each time step, it is necessary to check if a particle crossed a wall or not in order to introduce the architecture of the building in the filter. An approach to reduce this computation complexity is to limit the space the particles need to explore. Another representation for the building is a graph. These sets of edges and nodes make the skeleton of the building. Constraining the particles to move on this representation of the building is really interesting, as it is not necessary to check if particles crossed a wall or not.

The particle filter tries to estimate the probability distribution $\Pr\left[X_k \middle| Z_{0:k}\right]$ where X_k is

the state vector of the device at the time step k, and $Z_{0:k}$ is the set of collected measurements until the $(k+1)^{th}$ measurement. When the number of particles $\left(\text{positions } x_k^i, \text{ weight } \omega_k^i\right)$ is high, the discrete probability density function of presence can be assimilated to:

$$\Pr\left[X_k \left| Z_{0:k} \right.\right] = \sum_{i=1}^{N_s} \omega_k^i \cdot \delta\left(X_k - X_k^i\right) \quad (1)$$

This filter comprises several steps:

- Prediction: During this step, the particles propagate within the building given an evolution law that assigns a new position for each particle with an acceleration governed by a random process. New positions for all the particles are predicted.
- Correction: When a measurement (n-uplet of RSS) is available, it must be taken into account to correct the weight of the particles in order to approximate $\Pr\left[X_k \left| Z_{0:k} \right.\right]$. As the measurement is signal strength and given that particles are characterized by their position, the RSS n-uplet must be transformed into a position. The mapping between the position and the signal strength is performed thanks to the empirical database. Then it is possible to estimate $\Pr\left[Z_k \left| X_k \right.\right]$. Once defined all the necessary probabilities to update the weight of a particle, it is just needed to combine them to find the new posterior distribution.
- Update of the weights: The weight update equation is the one used in [4][5]. To obtain the posterior density function, it is necessary to normalize those weights. After a few iterations, when too many particles crossed a wall, just a few particles will be kept alive (particles with a non zero weight). To avoid having just one remaining particle, a re-sampling step is triggered.
- Re-sampling: The re-sampling step is a critical point for the filter. The basic idea behind the re-sampling step is to move the particles that have a too low weight, in the area of the map where the highest weights are. This leads to a loss of diversity because many samples will be repeated. Various re-sampling algorithms are proposed in [7]. We chose the simple SIS (Sequential Importance Sampling strategy).

2.2.1 The Voronoi diagram

The Voronoi diagram [8] has been used for a long time in the robotics community to model the environment in which a device is evolving. The Voronoi diagram is a set of edges that are equidistant to all the walls. The first stage is to automatically design this Voronoi diagram from a bitmap picture. A routine has been written to perform this task (figure 1). With such a representation it is possible to limit the moves of the particles. Now they are constrained to move on the edges of the oriented graph. This reduces the processing cost at each time step. There is no need to check if a particle crossed a wall or not. As they have a reduced area to explore, it is possible to cut down the number of particles. In our

simulations, only 200 particles were used to track the device. Indeed, the particles move on a graph which is a one dimensional space, whereas in the previous case, the particles were moving in a two dimensional space.

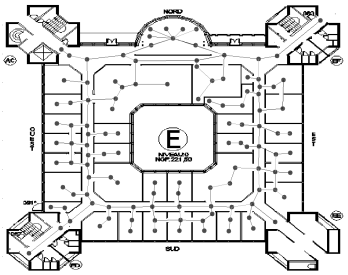


Figure 1: Voronoi diagram for a building (set of edges and nodes)

2.3 Experiments

To experiment with all those techniques and estimate their capabilities and accuracy to localize a device, a demonstrator has been built. It is made of a set of four 802.11g Linksys WAP54g APs placed at each corner of the 35×35 m building. The mobile device (PDA) is evolving in an indoor office environment. Both, a laptop and a Compaq iPAQ 4700 PDA were used for the measurements. The database is built with one measurement in each room, and a measurement every two meters in the corridor. The single floor problem is considered. The criterion to define the error is the mean error over a trip in the building. A walk around the building was made for the test. Some real measurements were collected along this path and then reused to estimate the performances of the positioning technique. Here, the measurements frequency is 3.33 Hz and the handheld device computes by itself its own position.

Obtained results show that using the raw fingerprinting it is not possible to recognize the path followed by the mobile moving across the building. As expected, it is necessary to filter information over the time to be able to obtain a coherent trajectory. The particle filter (with 200 particles) constrained on Voronoi diagram has been used to find the trajectory of the mobile. The estimated trajectories along the corridor are shown on figure 2.

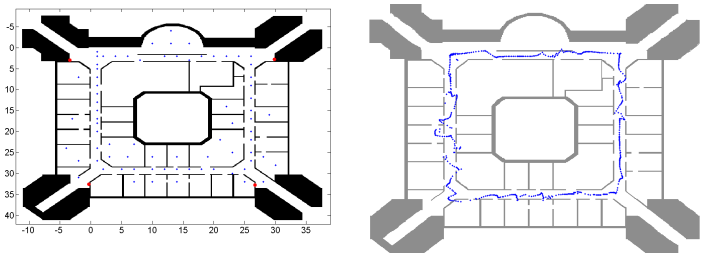


Figure 2 : Trajectory obtained with the raw fingerprinting (left) and with particle filter constrained on a Voronoi diagram (right)

It can be seen that the estimated trajectory using particle filter constrained over Voronoi diagram fits the real one. After some few time steps, the filter starts tracking the device correctly. The obtained results show an achievable accuracy of less than 2 m of error for the 66% of the cases with a low infrastructure. Increasing the density of APs improves the performance, however such deployment does not appear to be realistic. Hence, the particle filter constrained on a Voronoi diagram appears to be a good trade off between complexity (computation time of a measurement) and performance, as the performance of this filter are similar to the one achieved with the particle filter with particles freely moving.

3 TOA with IEEE 802.11 MAC frames

3.1 Introduction

The research challenge corresponds to achieve an indoor location system capable to provide accurate positioning using the existing WLAN infrastructure and devices with minor changes, avoiding the need of synchronization between APs and long manual system pre-calibrations (i.e setup of the fingerprinting database), while presenting robustness to environmental changes (i.e. furniture reorganization). Following this direction, a new WLAN location technique is presented, which can be divided into the ranging (distance estimation) and the positioning subsystem. The former estimates the distance between the Mobile Terminal (MT) and the AP from TOA estimation and the latter calculates the MT position using the distances estimated from the MT to 3 APs and the APs' known positions.

Several contributions existed in the scope of the proposed technique, but none of them fulfilled the degree of desired accuracy, simplicity and flexibility. In [9], a new approach is proposed to ranging in IEEE 802.11, without the requirement of initial synchronization between transmitters and receivers. Ranging is achieved by using a high precision timer in order to measure TDOA from two GRP (Geolocation Reference Point). The authors also propose to take advantage of the IEEE 802.11 data link frames for measuring TOA (time-of-arrival), but they do not give more insight to this matter. In [10], a system which can estimate TOA using IEEE 802.11 link layer frames is proposed, but the RTS (Request-to-Send)/CTS (Clear-to-Send) mechanism is required. In [11], a method to estimate TOA between WLAN nodes without using extra hardware is presented, but the achieved accuracy (error of 8 metres) is not enough for most safety related applications.

3.2 Ranging system

3.2.1 Round Trip Time (*RTT*) estimation

TOA is estimated from *RTT* measurements in order to avoid the need to synchronize the MT with the APs. *RTT* is the time a signal takes to travel from a transmitter to a receiver and back again, in our case from a MT to a fixed AP. As can be seen in figure 3, we estimate the *RTT* by measuring the time elapsed between two consecutive frames under the

IEEE 802.11 standard: a link layer data frame sent by the transmitter (it is the MT) and the reception of the correspondent link layer acknowledgement (ACK) from the receiver (it is the AP). Other link layer frames would be also suitable [10].

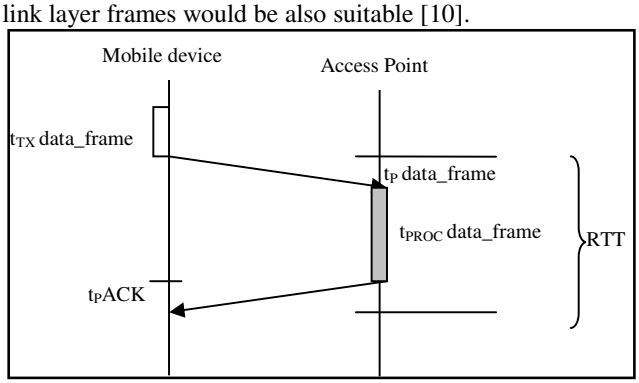


Figure 3: RTT measurement using IEEE 802.11 data/ACK frames

As the overall (i.e. propagation plus processing) *RTT* is expected to be in the order of microseconds, measuring it with software as in [11] leads to a significant lack of accuracy. Therefore, we propose to measure the *RTT* through a simple hardware module that starts counting cycles of the built-in 44 MHz clock from the WLAN card when it detects the end of transmission of a data frame, and it stops when the corresponding ACK frame arrives**.** Then it sends its value (i.e. slotted in 44 MHz periods) to the laptop PC. A lab prototype implementing this has been build, based on a laptop with an IEEE 802.11b PCMCIA card and the additional connected hardware module.

The *RTT* is time-variant due to constraints such as the variability of the radio channel multipath [12], the 44 MHz clock quantification errors [11], delays due to the electronics of the hardware module and the relative clock drift. If we only considered the quantification errors, a distance estimation error of 7 m should be present. In order to mitigate these errors this paper proposes to deal with *RTT* as a random variable: performing several (n) *RTT* measurements (i.e. samples) and using a proper *RTT* estimator over the *RTT* samples. The chosen *RTT* estimator was the average *RTT* value (η, measured in number of clock cycles) obtained from all the measurements, since among all tested choices this value provided the best *RTT* estimation. Other choices, such as the half range *RTT*, the *RTT* mode, the average of n minimum *RTT* values and η- β times the standard deviation were also tested, but they did provide lower accuracy and are not reported.

3.2.2 Distance estimation

First, a *RTT* estimation at zero distance between the MT and the AP is obtained (the propagation times t_p is zero), in order to calibrate the processing time in the AP. The figure obtained is assumed to be the $t_{proc\ data\_frame}$ part in Fig. 1, so that it can be used as an offset for measurements at a non-zero distance. Consequently, by applying the offset obtained, it

is possible to find the ΔRTT , it is the pure propagation time of the *RTT*:

$$\Delta RTT = RTT_a - RTT_0 \tag{2}$$

Once the ΔRTT is calculated -and being aware that a 44 MHz clock was used for the measurements and that the average RTT is used as estimator (η, measured in number of clock cycles)- the distance *d* (in meters) between the transmitter and receiver can be obtained as:

$$d = ((\eta_a - \eta_0) \cdot 3 \cdot 10^8)/(2 \cdot 44 \cdot 10^6) \tag{3}$$

3.2.2.1 Empirical coefficient

During the development process, it was observed that all the distances estimated were longer than the actual distances; therefore, the estimated distance had to be divided by an empirical coefficient to correct the estimated value. This coefficient is justified by the different sources of error commented before, which can increase the theoretical expected *RTT*. To estimate that coefficient, linear regression lines were traced for several distances relating the estimated distance obtained following the method described above with the actual distance. The obtained coefficient was *k*=0.694. Therefore the corrected formula for calculating the distance is:

$$d = ((\eta_a - \eta_0) \cdot 3 \cdot 10^8 \cdot k)/(2 \cdot 44 \cdot 10^6) \tag{4}$$

3.2.3 Experimental Test Bed and Measurements

The first experimental test bed consists of several distance estimations in the laboratory (indoors) in LOS situations between the lab prototype and the AP, for distances from 0 to 30 meters. The obtained mean distance estimation error taking into account all the tested distances was 0.81 meters. In a second set of measurements, the probability distribution of the distances estimated by the ranging system was obtained. This set consists of 450 distance estimations (450*300 *RTT* measurements) at a fixed distance of 10 metres, after the initial calibration at 0 metres. Ideally, all the distances measured should be 10 metres; however, due to several error sources, the ranging system obtains distances from 8.80 metres to 12.80 metres. The known probability distribution that best fitted it was found to be a Gaussian distribution, with $\mu = actual\_dist + 1.12$ and $\sigma = 0.84$.

3.3 Positioning System

3.3.1 Introduction

The MT position can be estimated once the distance estimations from a set of AP are obtained and the APs coordinates are known. The simplest option is to use a pure

triangulation algorithm, but higher accuracy can be achieved if tracking is applied, because it takes advantage of the past trajectory followed by the MT. Specifically, a Kalman-based tracking algorithm has been designed due to its simplicity and potential performance features. For a detailed description of the Kalman filter see [13] and [14].

3.3.2 Experimental Test Bed: Simulations

Simulations have been carried out in order to evaluate the performance of the positioning system using the Kalman-based approach. Furthermore, the Non Linear Least Squares (Newton) trilateration algorithm has also been implemented in order to evaluate the advantage of tracking results versus pure positioning techniques. For this evaluation it was desired to obtain the Cumulative Distribution Function (CDF) of the absolute positioning error. The observables that feed the filter (i.e. in the correction step) on every position estimate correspond to the distance estimations from the MT to the three nearest APs, using ranging statistical distribution presented before. A large number of routes (5000) with bad GDOP zones and probable changes of direction were generated following a motion model as similar as possible to a real behaviour of a pedestrian. The scenario is composed by a squared area of 50x50 m^2 with an AP in every corner. The positioning step T is set to 1 second.

3.3.3 Results

The obtained CDF of the absolute positioning error for the algorithms shows that the Kalman-based algorithm provides a high accuracy of less than 0.9 m. of absolute positioning error for the 66% of the cases (one sigma), and less than 1.4m. for the 90%. Comparing it with Newton, the improvement seems to be noticeable, because the latter provides 1.2 metres and 1.8 metres for 66% and 90% respectively. Figure 4 shows an interval of one of the generated MT's trajectory and the estimated ones obtained with Newton and the Kalman-based algorithms. It can be easily appreciated that the later provides an erratic path whereas the former is able to achieve a smoothed trajectory very similar to the actual one.

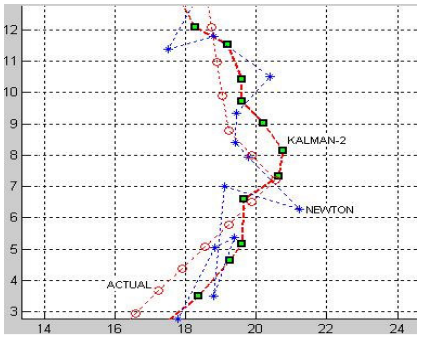


Figure 4: Actual and estimated trajectories

4 Other technologies

4.1 Ultra Wide Band (UWB)

The work on UWB-based positioning focused on the analysis of the potential ranging and positioning accuracy of future low data rate UWB systems compliant to the IEEE 802.15.4a standard, currently under development. The analysis took into account the characteristics of the UWB propagation channel (in terms of both communication range and impact on the ranging accuracy) as well as the MAC strategy to be adopted in the 802.15.4a standard, based on Aloha, and the impact of Multi User Interference.

Two different scenarios were selected for this analysis in order to represent the different application scenarios expected to be served by the new IEEE 802.15.4a standard. The first scenario was characterized by a centralized controller determining the position of fixed or mobile nodes based on the distance estimations provided by a set of fixed reference nodes. A Time Difference Of Arrival approach adopting a Least Square Error minimization was used for estimating the position of the terminals at the central controllers. An indoor environment characterized by both LOS and NLOS links was considered. Simulation results show that the positioning error is potentially very low in the case of LOS links between nodes, and remains acceptable even in presence of a significant percentage of NLOS links.

The second scenario addressed the case when no external infrastructure is available, and relative position information must be built from scratch within the network. The scenario was characterized by a network of terminals that build a coordinate system exchanging distance and position information by means of a distributed algorithm derived from the Self Positioning Algorithm (SPA) [15]. In this scenario the results indicated that, for a network with high enough terminal density, a distributed protocol combined with an IEEE 802.15.4a UWB physical layer can potentially provide accurate position information even in absence of any external infrastructure, despite the potentially high MUI interference caused by the strong signalling overhead in the construction of the common coordinate system required by the SPA algorithm. Further information on the results of UWB research activity within LIAISON can be found in [16], [17] and [18].

4.2 Inertial Navigation Systems

Inertial Navigation Systems (INS) are commonly used in the naval and aviation fields. While pedestrian navigation can be based on the same underlying principles, i.e. measure of accelerations and angular velocities, the quality of the sensors employed differs quite significantly from those used in “traditional” inertial systems. Due to constraints on ergonomics (weight and size), power consumption and price, the sensors used in pedestrian navigation are based on Micro-Electro-Mechanical Systems (MEMS) technology [19]. The Liaison research activities in this domain of MEMS based location has focused on two primary axes:

- Development of algorithms for real-time implementation to detect and characterize human physical activities, which include both body postures (lying, sitting and standing) and body displacement (distance travelled and azimuth of travel).
- Coupling of MEMS derived body displacement with absolute positioning information provided by other technologies (e.g. A-GNSS or WiFi based).

With respect to the first point, a novel approach in the context of pedestrian navigation has been pursued that consists on placing sensors in different parts of the human body, specifically the trunk, thigh and shank. With this architecture it is possible to determine the real posture of a pedestrian. This information is not only useful to infer about his safety condition, but also to adjust the navigation algorithms to certain specific movements of the professional users (e.g. firemen, which can crawl or walk squatted) [18]. In what concerns the quantification of the distance travelled by a pedestrian, an original approach based on Fuzzy logic classifier has been developed to improve the detection and identification of the type of walking (e.g. forward/backward, stairs climbing/descending). As for determination of the azimuth of travel, a new algorithm has also been implemented that fuses the information provided by all three types of sensors available (gyroscopes, accelerometers and magnetometers), being capable of mitigating the magnetic disturbances that induce significant errors in orientation.

To validate these algorithms and assess their performance, several tests have been conducted. For an indoor heterogeneous path travel of more than 500 meters, including stairs climbing and descending, positioning errors of less than 20 meters were always observed in dead reckoning mode only.

Regarding the second research axis pursued under Liaison, work has been carried out to hybridize MEMS based positioning with other positioning technologies, namely A-GNSS and WiFi TOA/Fingerprinting [22]. Besides increased availability of the overall system, this approach allows the correction of certain systematic errors on the MEMS side (i.e. step length and orientation errors), improving positioning accuracy and robustness [20][21].

5 Conclusion

Indoor location with WiFi allows using the existing infrastructure and devices widely deployed in buildings such as airports, train stations, hotels, etc. The two approaches presented in this paper provide a good accuracy. UWB (in adhoc mode) and INS can be used in emergency scenarios, e.g. fire fighting, when infrastructure is disconnected.

References

[1] A; Motley and J. Keenan, "Personal communication radio coverage in buildings at 900 MHz and 1700 MHz", Electronics Letter, vol. 24, June 1988

[2] R. Vaughan and J. B. Andersen, Channels, propagation and antennas for mobile communications. Electromagnetic Waves Series 50, The IEE, London, United Kingdom, 2003

[3] A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment", IEEE Wireless Communications, Oct. 2003

[4] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system", Proceedings IEEE Infocom 2000, Tel Aviv, Israel, col. 2, pp. 775-784, Mar. 2000.

[5] R. Battiti, T. L. Nhat and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANSs", tech. rep, Dept. of Information and Communication Technology, Univ. of Trento, Feb. 2002

[6] A. Hatami and K. Pahlavan, "A comparative performance evaluation of RSS-based positioning algorithms used in WLAN networks", IEEE Wireless Communications and Networking Conference, 2005

[7] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for on-line non-linear/nongaussian bayesian tracking", IEEE Transactions on Signal Processing, vol. 50, Feb. 2002.

[8] L. Liao, D. Fox, J. Hightower, H. Krautz, D. Schultz, "Voronoi tracking: location estimation using sparse and noisy sensor data", Proc. Of the International Conference on Intelligent Robots and Systems (IROS, IEEE/RSJ, 2003.

[9] X. Li; K. Pahlavan, M. Latva-aho, M. Ylianttila, "Comparison of Indoor Geolocation Methods in DSSS and OFDM Wireless LAN Systems", *IEEE VTS-Fall VTC 2000*. 52nd, Volume 6, 24-28 Sept. 2000, pp. 3015-3020.

[10] D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, M. Martorana, "Mobile Ranging Using Low-accuracy Clocks", *IEEE Transactions on Microwave Theory and Techniques*, Volume 48, Issue 6, June 2000, pp.951-958.

[11] A. Günther, C. Hoene, "Measuring Round Trip Times to Determine the Distance Between WLAN Nodes", *Networking 2005*, pp. 768-779

[12] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavraki, "On the Feasibility of Using Wireless Ethernet for Indoor Localization", *IEEE Transactions On Robotics And Automation*, Vol. 20, No. 3, pp.555-559, June 2004.

[13] R. Kalman, "A new approach to linear filtering and prediction problems", Trans. ASME, J. Basic Eng. 82D, pp. 35--45, 1960.

[14] G. Welch and G. Bishop. "An introduction to the kalman filter". Technical Report TR 95-041, University of North Carolina at Chapel Hill, 1995

[15] S. Capkun, M. Hamdi and J. P. Hubaux, "GPS-free positioning in mobile Ad-Hoc networks," Hawaii International Conference On System Sciences, HICSS-34 January 3-6, 2001 Outrigger Wailea Resort, pp. 3481 - 3490.

[16] R. Cardinali, L. De Nardis, P. Lombardo and M.-G. Di Benedetto, "UWB Ranging Accuracy in High and Low Data Rate Applications," Special Issue on UWB, IEEE Transactions on Microwave Theory and Techniques, Volume 54, Issue 4, April 2006, pp. 1865 - 1875.

[17] L. De Nardis and M.-G. Di Benedetto, "Positioning accuracy in Ultra Wide Band Low Data Rate networks of uncoordinated terminals," IEEE International Conference on UWB 2006 (ICUWB 2006), pp. 611-616, Sept. 2006, Waltham, MA, USA

[18] LIAISON Project, Deliverable D054 "Solution Research Results Data Package [Iss. 02]," July 2006.

[19] Q. Ladetto and B. Merminod, "In Step with INS – Nvaigation for the Blind, Tracking Emergency Crews", in GPS World, October 2002.

[20] Q. Ladetto, "On foot navigation : continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering", in ION GPS 2000, 2000, Salt Lake City.

[21] Q. Ladetto et al., "Human Walking Analysis Assisted by DGPS", in GNSS 2000, Edinburgh.

[22] V. Renaudin, O. Yalak, and P. Tomé. Hybridization of MEMS and Assisted GPS for Pedestrian Navigation. Inside GNSS, January/February: 34-42, 2007.

Improving Unsynchronized MAC Mechanisms in Wireless Sensor Networks

Philipp Hurni and Torsten Braun

Institute of Computer Science and Applied Mathematics, University of Bern
hurni, braun@iam.unibe.ch

Abstract. Energy-saving MAC-layer mechanisms in wireless sensor network nodes generally consist in periodic switching of a low-power wireless transceiver between an energy saving sleep mode and the costly operation modes receive and transmit. Many approaches aim to synchronize the state changes of the nodes in the network and introduce mechanisms to let the nodes synchronously wake up at designated points of time in order to exchange pending traffic. Synchronized schemes are difficult to achieve, especially over multiple hops, and introducing control messages for global or clusterwise synchronization can be a costly issue.
This paper examines improvements and optimizations on recently proposed power saving MAC protocols based on asynchronous wake-up patterns and wake-up announcements, and tests them out in a wireless sensor network integrated with an ad hoc on demand routing protocol.

1 Introduction

The design of energy efficient medium access protocols is a challenging task. It consists in finding means to use the wireless transceiver only in an *on demand* manner. In wireless sensor networks, this task is of crucial importance, as the transceiver hardware is accountable for a major part of a node's energy consumption. To save energy, the transceivers have to be switched into a low-power sleep state for a maximum amount of time, yet still maintaining connectivity to the neighboring nodes, in order to keep the network operable. The major part of many power saving mechanisms consists in introducing central or distributed synchronization and periodic switching between a sleep state and a wake state. Such synchronization measures however always cause new overhead, especially when applying multi-hop synchronization schemes. In low-traffic scenarios, periodic control message overhead can exceed the energy spent for the actual payload. Recent publications therefore proposed variants of unsynchronized power saving mechanisms.
In this paper, we lean on the basic concepts of a fully unsynchronized power saving mechanism introduced in [1] and [2], apply it to a wireless sensor network scenario and integrate principles of the WiseMAC [3] protocol. The following sections first introduce into previous work on unsynchronized power saving mechanisms, then suggest and discuss improvements concerning the wake-up

patterns and quantify the simulated efficiency gains in different networking scenarios. The novelty to be discussed lies in the introduction a moving preamble sampling period besides the fixed sampling period of WiseMAC, through which a deterministic wake-up scheme with fewer collision and fairness problems can be obtained.

2 Related Work

2.1 S-MAC

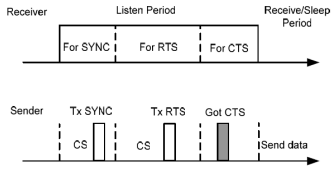


Fig. 1: S-MAC duty cycle

The S-MAC [5] protocol aims to synchronize the wake-up patterns of the nodes. It aims to let the nodes simultaneously wake up and fall back to sleep. S-MAC follows a *virtual clustering* approach to synchronize the nodes to a common wake-up scheme with a slotted structure. By regularly broadcasting SYNC packets at the beginning of a slot, neighboring nodes can adjust their clocks to the latest SYNC packet in order to correct relative clock drifts.
In a bootstrapping phase, nodes listen for incoming SYNC packets in order to join the ad-hoc network, and join a virtual synchronization cluster. When hearing no SYNC's, a node starts alternating in its wake-up pattern and propagates its schedule with SYNC messages. A problem of the virtual clustering arises when several clusters evolve. Bordering nodes in-between two clusters have to adopt the wake-patterns of both clusters, which imposes twice the duty cycles to these nodes. An S-MAC slot consists in a listen interval and a sleep interval. The listen interval is fragmented into a synchronization window to exchange SYNC messages, and a second and third window dedicated to RTS-CTS exchange. Nodes with receiving a RTS traffic announcement will clear the channel with a CTS respective window, and stay awake during the sleep phase, whereas all other nodes will go back to sleep.
The slot length and duty cycle must be set in a fixed manner, which severely restrains latency and maximal throughput. This can be disadvanteageous, as traffic can often be of bursty nature and the rate of traffic can vary over time.

2.2 T-MAC

The static duty cycle duration of S-MAC results in high latency and lower throughput, especially when varying the load level. Timeout-MAC (T-MAC)

[6] is proposed to enhance S-MAC unter variable load, and introduces an adaptive duty cycle. In T-MAC, the listen interval ends when no activation event has occurred for a given time treshold TA. An activation event may be the sensing of any communication on the radio, the end-to-end transmission of a node's data transmission, overhearing a neighbor's RTS or CTS which may announce traffic destined to itself.
One drawback of T-MACs adaptive time-out policy is that nodes often go to sleep too early. T-MAC proposes a *Future Request to Send* (FRTS) control message to alleviate this so-called early-sleeping problem, to keep neighboring nodes awake for later data exchange when having lost the contention.

2.3 WiseMAC

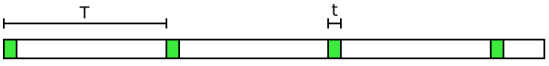


Fig. 2: WiseMAC operating with fixed cycle duration T and short medium sampling interval t

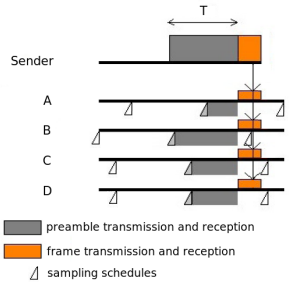


Fig. 3: WiseMAC broadcasting

The WiseMAC [3] protocol's wake-up scheme consists of simple periodic wake-up's and duty cycles of only a few percent in order to sense the carrier for a preamble signal, as depicted in Figure 2. A preamble precedes each data packet for alerting the receiving node not to go to the sleep state upon reception of the frame. As in [1], all the nodes in a network sample the medium with a common basic cycle duration T, but their offsets are independent and left unsynchronized. If a node receives the preamble signal when waking up and sampling the medium, it continues to listen until it receives a data packet or the medium becomes idle again. If a node does not know its neighbors' wake pattern yet, it sends a preamble of duration T, in order to reach the sampling interval of the neighboring node. After successful frame reception, the receiver node piggybacks its wake-up pattern in the acknowledgement message, which is then kept in a table containing the neighboring nodes' relative schedule offset from the own wake pattern. Based on this table, a node can determine the next wake-up of all its respective neigbors, and minimize the preamble length for all upcoming frames to the maximun clock drift that the two involved node's clocks may have developed during the time since the last wake pattern update.
In the WiseMAC protocol, the carrier sensing range is chosen to be larger than the transmission range in order to avoid collisions and mitigate the hidden node problem. For broadcasting, WiseMAC proposes to prepend a preamble of dura-

tion T on every broadcast message, in order to first alert every neighboring node for the upcoming transmission, and finally transmit the frame - exactly as it is done when sending a frame to a node when not knowing its sampling pattern. As illustrated in Figure 3, this broadcasting scheme uses a lot of energy only for sending and receiving the long preamble, whereas the actual data transmission may be much shorter.

2.4 Unsynchronized Power Saving Mechanism with Fixed and Random Interval

The mechanism proposed in [1] and analyzed in a static multihop wireless ad hoc network environment in [2] defines two wake and two sleep periods during one basic cycle duration T, as depicted in Figure 4.

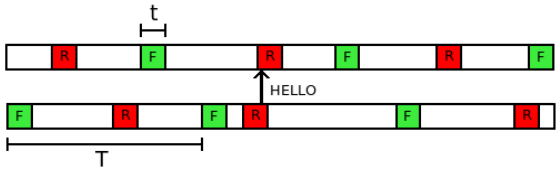


Fig. 4: Unsynchronized with Fixed and Random Interval

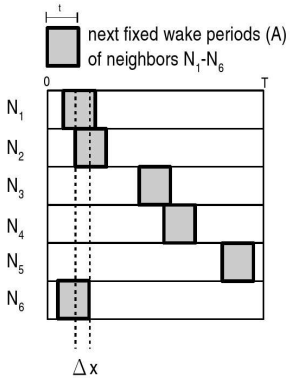


Fig. 5: Announced fixed wake periods in an intersection table

Nodes strictly alternate between a fixed wake period (F) and a random wake period (R). Each of the wake periods shall be of the same duration t. The start of the random wake period (R) is uniformly distributed between the end of the fixed wake period (F) and the start of the next one. All nodes operate with the same basic cycle duration T, although remaining unsynchronized, and switch between wake and sleep states in their individual wake-up pattern. Nodes all operate with the same wake ratio $W = 2t/T$.
The fixed wake period (F) enables a node aiming to contact any neighboring node, if its periodically occurring fixed wake period pattern is known.
However, if there is no intersection between the fixed wake periods of the sender and the neighbor, it may never learn about its presence. This motivates the choice for the random secondary wake period (R). It ensures that two nodes with disjoint wake-up pattern will sooner or later be awake at the same time and therefore be able to exchange announcements about their own wake period. By receiving these, nodes will be capable to reach all neighboring nodes during their fixed wake period (F).

As examined in [2], this wake-up scheme can easily be applied to multi-hop wireless ad hoc networks and reactive routing schemes. In order to efficiently disseminate broadcast messages, one can exploit the information about the next soonest wake-ups of each node's neighboring nodes.
By figuring out the best instant for sending and forwarding a broadcast message, [2] suggests to make optimal use of the so-called wireless multicast advantage. A node intending to broadcast a message can figure out the best instant to forward the message. The best instant shall be the instant during the next basic cycle T when the largest subset of the neighboring nodes is awake, aiming to transmit the message during some neighbor's intersections, if there are.
Figure 5 depicts the concept to search the best instant. The node calculates the best instant for broadcasting a messaget to be in-between Δx. The aim of the broadcast is therefore not to reach all of the neighbors, but only the largest possible count of neigbors with each attempt, as it is done in probabilistic broadcasting techniques, which furthermore alleviates the broadcast storm problem.
Using this technique, and taking the two best instants for rebroadcasting a route request of an on-demand routing protocol, the success ratio reached 97% even for the very low wake ratio of 4%. By rebroadcasting each message twice in every node, the disadvantage of the unsynchronized wake-pattern in regard of broadcasting becomes negligible, when considering the efforts that would otherwise be necessary to achieve a rigidly synchronized wake-pattern.

3 Simulation Environment

In all upcoming simulation scenarios, we used the OMNeT++ Network Simulator [8]. We made use of the Mobility Framework from TU Berlin [9], a framework to support simulations of wireless and mobile networks within OMNeT++. This framework incorporates a sophisticated transmission model which is based on calculation of SNR (Signal-to-Noise Ratio) and SNIR (Signal-to-Noise-and-Interference Ratio) values according to a restricted free space propagation model. This model takes transmitter power, distance, wavelength and path loss coefficient of signal dispersion into account.
The radio propagation model does not take multipath propagation or doppler effects into account, but allows to adjust the path loss coefficient α. Recent examinations of the signal attenuation in IEEE 802.11-based networks [4] conclude that a path loss coefficient between 3 and 4 is most suitable to model wireless propagation in office buildings and outdoor areas. Many sensor network simulations incorporate a path loss of 3.5 and more for wireless sensor network scenarios. We therefore sticked also to the same value of $\alpha = 3.5$. The energy consumption model is based on the amount of energy that is used by the transceiver unit. We do not take processing costs of the CPU into account. Each node's energy consumption is calculated in respect of the time and input current that the nodes spend in the respective operation modes idle/recv, transmit and sleep. Furthermore, state transition delays are incorporated to model the state

transition costs.
It is planned to later port the WiseMAC and the mechanisms described below to the Embedded Sensor Boards (ESB) of ScatterWeb [10]. The Simulation parameters are therefore tailored to model the ESB node's hardware characteristics. The ESB is equipped with a low power micro controller, various sensors, and a tr1001 transceiver module.

Simulation Parameters	
nodes	90 (uniform distribution)
area	300m × 300m
communication range	50m
carrier sensing range	100m
bitrate	19,2 Kbps
carrier frequency	868 MHz
transmitter power	0.1 mW
SNR threshold	4 dB
path-loss coefficient α	3.5
MAC & routing header	80 bit
payload	80 bit
sleep current	5 μW
transmit current	12 mW
idle/recv current	4.5 mW
recv to sleep transition delay	10 μs
recv to send transition delay	12 μs
send to recv transition delay	518 μs

4 Optimization of the WiseMAC Broadcast

With WiseMAC, broadcast transmissions must be of the duration of the sampling period to wake up and reach every node in range. As illustrated in Figure 3, this broadcasting scheme wastes a lot of energy for sending and receiving the long preamble, whereas the actual data transmission is much shorter.
In [2], this problem is studied when applying the unsynchronized wake-up pattern discussed in [1], which shares many similarities with WiseMAC. Both mechanisms propose to renounce on any global or clusterwise synchronization scheme, and only exchange information about the nodes' schedule offsets, and all nodes' wake patterns operate with a basic cycle duration T. We ported the k-best-instants heuristic of [2] into the WiseMAC mechanism of periodic preamble sampling with $T = 250ms$ and a 5% duty cycle to sense the carrier for pending traffic. An even lower duty cycle might be possible on some sensor hardware testbeds, but due to impreciseness and unpredictable behaviour of the state transitions, 5% should be an appropriate and realistic choice.
Broadcasting is of high importance when dealing with on-demand routing protocols such as AODV [7] or DSR. Nodes aiming to transmit a packet have to search a path to the destination by initiating costly Route Request floods. We

chose AODV as a well-established, efficient routing protocol, because its one-hop paradigm fits well to WiseMAC with its schedule offset table of the one-hop neighbors, and because AODV neglects to transmit and store the full routing information between two endpoints. With AODV, the route knowledge itself is distributed in the network, which makes sense in a resource-constrained wireless sensor network.
We tested out the performance of the upper schemes in an AODV route establishment scenario where every node in the network aims to find a route to the sink. In the following, the nodes first go through a neighborhood discovery process of 5 seconds during which they find their respective neighbors by sending a few HELLO messages using the original WiseMAC broadcast mechanism. After 1 minute, the first node emits a AODV route request message for the sink as it wants to start reporting data. After receiving a route response, the packet is forwarded hop by hop to the source by unicast. In intervals of 5 seconds later, one node after the other does the same, until every node has found a route to the sink. After 500s, the simulation is stopped, and the total energy consumption of all nodes calculated and summed up.

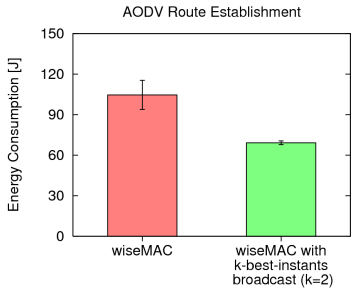


Fig. 6: Improved Broadcast scheme for WiseMAC

With both broadcasting techniques, every node managed to find a path to the sink and transmit the unicast packet. As we can see in Figure 6, the k-best-instants approach already leads to an efficiency gain of approximately 40% in this simple AODV Route Establishment scenario. The performance gain of the k-best-instants broadcasting technique weights as much as broadcasting and flooding mechanisms are used in the scenario. For example in an application scenario where queries are flooded to the nodes, the energy consumption of the approach operating with the WiseMAC broadcasting technique can be vastly improved by applying the k-best-intersections-broadcasting scheme.

5 Optimization of the WiseMAC periodic wake-up scheme

WiseMAC [3] proposes to switch the transceiver between receive and sleep state in a fixed periodic manner, and incorporating no synchronization between the

nodes other than learning each others sampling patterns. [3] argues that systematic collisions that would have been introduced through synchronization are mitigated using a probabilistic medium reservation scheme. As depicted in Figure 7 every node has its own switching pattern. Once a node has been turned on, it starts alternating between the receive and the sleep state, which leads to uniformly distributed medium samplings of the nodes. Systematic overhearing, as it occurs in synchronized MAC protocols like SMAC [5] and TMAC [6], does only seldomly occur, as in most cases, the non-intersecting wake-up intervals of the nodes naturally lead to a so-called *probabilistic overhearing avoidance*.
This scheme however has also clear drawbacks: The static deployment of a simple fixed-period sampling pattern makes it *impossible* for nodes to learn about the presence of their local neighbors just by overhearing messages originated by them. Systematic and permanent overhearing is energy waste, but some limited and infrequent overhearing can be advantageous, especially in ad hoc networks. With wake-up schedules piggybacked to all MAC frames, nodes overhearing traffic of neighbors can always update their schedule offset table. With only one fixed period wakeup pattern, nodes need to rescan the local neighborhood periodically in order to discover neighboring nodes, either by using the WiseMAC broadcasting scheme or other techniques. One simple WiseMAC broadcast right after deployment does not guarantee that all nodes were reached. It is possible that the broadcast was interferred or has failed due to bit errors.

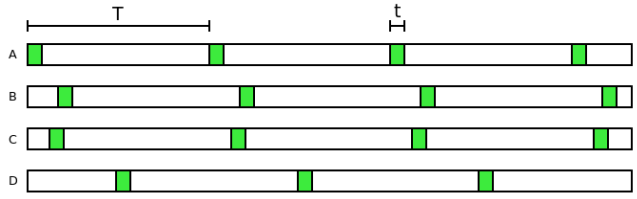


Fig. 7: WiseMAC nodes operating with fixed wake-up pattern

Furthermore, two nodes with nearly identical sampling patterns might systematically hinder each other from receiving messages destined to them. Consider node B and C in Figure 7, which share almost the same wakeup pattern. Assume that all node are at least in interference rage of each other.
Node C always slightly precedes the wakeup period of node B. If two respective neighbors A, D want to reach B and C, the transmission $A \rightarrow B$ will always be shadowed by the transmission $D \rightarrow C$, as node C always wakes up earlier. D will always be capable of sending the preamble and start transmitting the frame to C, whereas B will wake up, notice that there is a transmission going on that is not destined to itself and go back to the sleep state after the medium is idle again. A will have to wait until there is no message transfer to C such that it can finally transmit to B. This leads to a high latency for A's packets whenever there is traffic destined to C.
Such problems can have severe impact on the service properties for a large part of the nodes, especially if C and D are neuralgic spots in the sensor network

which have to forward data packets from whole subtrees.
Another drawback of the single periodic wake period occurs when applying the k-best-instants broadcasting scheme introduced in [2] to on demand route request querying, as the neighboring nodes will always consider the same nodes' intersections for rebroadcasting a frame and therefore stick to the same behaviour in every retry attempt. This is especially the case when there are bottlenecks in the network topology and is investigated furthermore in [2].
The WiseMAC fixed periodic wake-up mechanism can be improved in a quite simple manner. We can achieve a medium reservation scheme with similar properties but a better probabilistic overhearing avoidance and better medium utilization by keeping a fixed wake-up period and integrating a moving wake period in between two fixed wake periods. A node then strictly alternates between a fixed wake period and a moving wake period, similar to the mechanism proposed in [1].

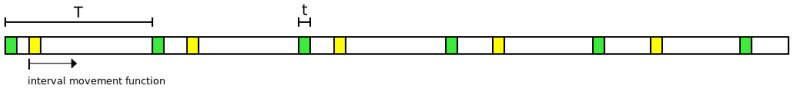


Fig. 8: fixed wake period (green) and moving wake period (yellow)

[1] proposed the choice of a fixed wake-up pattern and a random wake-up period in between to solve the problem that nearby nodes are possibly never detected due to non-intersecting wake patterns. We suggest a mechanism with the fixed wake period and the moving wake period, as shown in Figure 8, for the following reasons:

- The behaviour of the moving wake period is deterministic rather than random and follows a simple linear movement function, which is identical and predictable in every node. If a node needs to transmit a message to one of its neighbor, it first checks whether the neighbor's next fixed wake period or the next moving wake period is soonest.
  Using previously received wake-pattern annuncements, the node then determines the next soonest wake period of the neighbor, prepones a preamble to the frame and then awaits its neighbor's wake-up exactly as in WiseMAC.

- The problem that non-intersecting wake-up patterns could lead to nodes never discovering each other is also resolved. Sooner or later, nodes will overhear frames or acknowledgements, even from or to nodes with non-intersecting wake-up patterns. The moving wake interval ensures that - given some periodic low-rate traffic - this will happen within a limited amount of time.

- Using this wakeup pattern, the upper problem with the concurrent transmission between the nodes $A \rightarrow B$ and $D \rightarrow C$ is solved in an elegant manner. Consider the situation depicted in Figure 9:
  Node A aims to transmit a frame to node B at the instant indicated with the

upper arrow (i.e. it may have to forward the frame after receiving it during its own fixed wake period), and D aims for transmission of a frame to C at the instant indicated with the lower arrow. D will find the medium idle at the start of the next fixed wake period of node C and will transmit the packet. A will find the medium busy at the start of the next fixed wake period of B, and not access the channel. It will be able to wait for the next moving wake period of B, which does not intersect with the respective moving wake nor fixed wake period of C.
The movement function of the moving wake period leads to a floating of the node's wake periods over time. If there was only a fixed period, A would have to wait until D has no more packets to send. If D continously generates or forwards packets, the traffic that needs to be forwarded by B is blocked. This can lead to high delays for packets forwared by B, causes fairness problems and can even lead to buffer overflows at B.

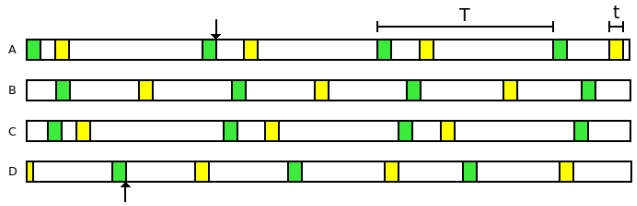


Fig. 9: Problem of concurrent transmissions

5.1 Node-to-Sink Periodic Traffic Scenario

We compared the approaches of the k-best-instants WiseMAC Scheme (original wakeup pattern but improved broadcasting technique) with the approach of the moving wake intervals described above. In case of the WiseMAC wake-up pattern, we chose $250ms$ as interval between two duty cycles. In case of the moving wake interval approach, we chose $500ms$ as wake-up interval between two fixed duty cycles. Like this, the expected value of the wake-up interval of both approaches equals 250ms, and the service characteristics of both approaches can be compared.
We chose the same networking setup as in the upper scenario, and let every node report data starting at $t = 60s$ with poisson traffic of increasing rate λ during $1h$. As we are only dealing with the route establishment in the beginning and the broadcasting scheme is the same in both approaches, the comparison mainly covers the properties of the unicast node-to-node acknowledged datagram service from the sources to the sink when applying the different wake-up schemes.

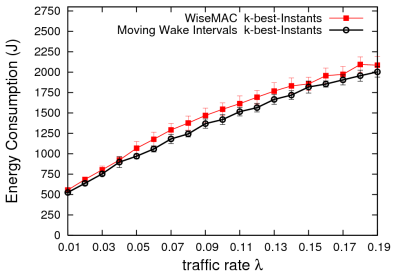


Fig. 10: Energy Consumption

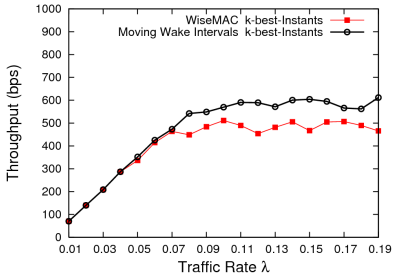


Fig. 11: Throughput

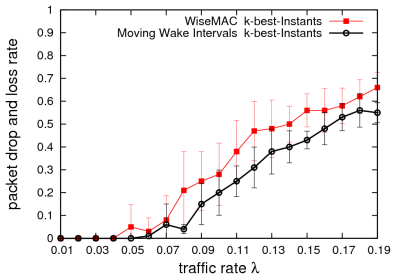


Fig. 12: Packet Drops and Losses

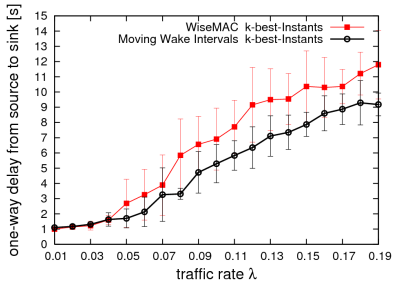


Fig. 13: One-Way-Delay

The moving wake interval approach leads to a slightly lower energy consumption and a better throughput with increasing traffic rate. The performance gains in regard of throughput and energy consumption are measurable, though remain below 20%.
It is insightful that the mechanism with the moving wake periods performs better only with increasing traffic. As long as there is not much traffic, the situation with the concurrent transmissions described above does not or only seldomly occur. With increasing traffic, congestion problems arise earlier with the fixed static wake-up pattern of WiseMAC.

5.2 Distributed Events Scenario

Similar results as in 5.1 can be observed in a distributed event scenario. We triggered events with poisson rate λ on a random position point on the same uniformly distributed topology. When a event happens at a certain position on the $300m \times 300m$ plane, each node in the vicinity of $50m$ starts reporting data with between 1 and 3 packets.
The results also show an improvement of the moving wake period mechanism in comparison with the fixed wake pattern, which is slowly increasing with increasing event rate.

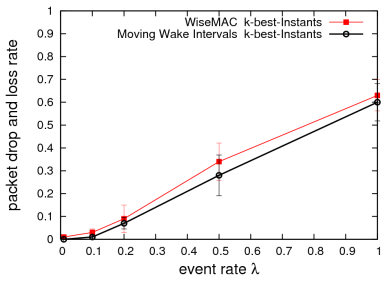


Fig. 14: Packet Drops and Losses

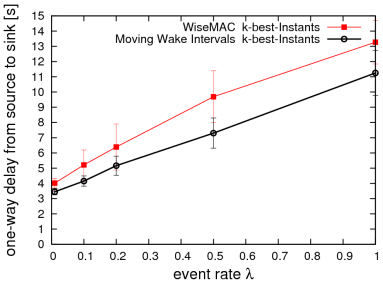


Fig. 15: One-Way-Delay

6 Conclusions

This paper combines features and ideas of previous work on unsynchronized MAC protocols for wireless sensor networks and finds performance optimizations in regard of energy efficiency, throughput, fairness and latency. It shows that mechanisms suggested in [1] and [2] can be applied to improve the yet very effective WiseMAC [3] power saving MAC protocol. Porting the k-best-instants broadcasting technique to a multihop wireless sensor network led to performance gains in comparison with the WiseMAC broadcasting scheme. We showed that the WiseMAC fixed periodic wake-up scheme can cause fairness problems with increasing traffic, which can be alleviated by adapting the wake-up scheme to a hybrid scheme with a fixed periodic wake-up and a moving wake-up.

References

1. Braun, T., Feeney, L. M.: *Power Saving in Wireless Ad hoc Networks without Synchronization*, 5th Scandinavian Workshop on Wireless Ad-hoc Networks, 2005
2. Hurni, Ph., Braun, T., Feeney, L.M.: *Simulation and Evaluation of Unsynchronized Power Saving Mechanisms in Wireless Ad Hoc Networks*, WWIC 2006
3. El-Hoiydi, A., Decotignie, J.-D.: *WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks*, ALGOSENSORS 2004
4. Faria, D.: *Modeling Signal Attenuation in IEEE 802.11 Wireless LANs* Technical, Stanford University, July 2005.
5. Ye, W., Heidemann, J., Estrin, D.: *An Energy Efficient MAC protocol for Wireless Sensor Networks.* 21st International Conference of the IEEE Computer and Communications Societies (Infocom), 2002.
6. Dam, T. V., Langendoen, K.: *An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks.* ACM Conference on Embedded Network Sensor Systems (SenSys), 2003.
7. Perkins, Ch., Belding-Royer, E.: *Ad hoc On-Demand Distance Vector (AODV) Routing*, IETF Internet draft RFC 3561, October 2003
8. Varga, A.: *The OMNeT++ Discrete Event Simulation System (http://www.omnetpp.org)*,
9. Mobility Framework for OMNeT++ (http://mobility-fw.sourceforge.net)
10. Scatterweb: *Platform for wireless sensor networks* (http://www.scatterweb.net)

A Middleware Approach to Configure Security in WSN

Peter Langendoerfer[1], Steffen Peter[1], Krzysztof Piotrowski[1], Renato Nunes[2], and Augusto Casaca[2]

[1] IHP, Im Technologiepark 25, 15236 Frankfurt (Oder), Germany
{langendoerfer|peter|piotrowski}@ihp-microelectronics.com
[2] INOV, Rua Alves Redol, 9 - 1000-029 Lisboa, Portugal
{renato.nunes|augusto.casaca}@inesc-id.pt

Abstract. Security configuration of standard systems is a tedious and error prone task. Doing this for WSN is even more complex due to the scarce resources of the sensor nodes. In order to simplify this task we propose a middleware architecture as well as a configuration tool. The main idea is that the configuration tool selects security providing modules such as appropriate cipher means. The choice is based on a detailed description of security needs of the application under development as well as on the description of the available security modules and sensor nodes. The middleware architecture supports configuration before and after deployment of the sensor nodes. It consist of an essential core that provides configuration features and an additional layer in which the security modules are clustered.

1 Introduction

It is obvious that security is an essential need in ubiquitous wireless computing and there are plenty of means that promise to secure Wireless Sensor Networks (WSNs). In WSNs obvious parameters like security strength are overshadowed by network structure, the number of available base stations, requirements in response times, frequency of security operations and especially by the energy consumption of cryptographic operations. All these parameters must be considered if security is supposed to be incorporated in the WSN. This is a very challenging task even for security experts, and programming of a WSN is already a complex task by itself.

Our approach to reduce complexity of the realization of an appropriate security level for a given WSN application is to provide a configurable and adaptive security middleware. The configuration of an initial set of security modules is done by our configuration kit (configKIT) before deployment of the application. With respect to security issues, the application programmer has to specify only which security functionality, e.g. data secrecy and authentication is needed by the application. In addition she has to provide some information on the sensor node configuration, e.g. processor, memory size etc. Based on this data the configKIT selects the appropriate security modules. The security modules come

with a self description concerning functionality and required resources for example. These modules are provided by the UbiSec&Sens project [16] into which our activities are also embedded. Adaptability ensures that security modules can be exchanged after deployment, e.g. if application needs have changed or if vulnerabilities of a certain module have been detected which require an update. This functionality is achieved by our modular architecture which separates core functionality needed for adaptability support from pure security functionalities.

The rest of this paper is structured as follows. Section 2 provides a short state of the art. The configuration tool is introduced in section 3. The following section discusses our middleware architecture. The paper concludes with a short summary and an outlook on further research steps.

2 State of the Art

Wireless sensor networks are mainly used to gather data about a certain environment, see for example [13]. Due to this focus also research in the middleware area has somewhat concentrated on supporting data storage and retrieval issues in WSNs. Some prominent approaches are tinyDB [10], Cougar [18] and Hood [17] to name just a few.

Some work towards flexible middleware for WSNs has already been done. These approaches try to provide application independent support to applications but are mainly focusing on communication issues in one form or another. In [19] authors introduce the concept of reconfigurability for middleware in pervasive computing. Here the major part, if not all components, of the middleware is located at a PDA device and the task of the middleware is merely discovery provision of available data. Authors of [15] propose an application independent scheme for defining groups of sensors to provide easy adaptability of a WSN to new applications. Here part of the adaptation logic is placed at the sensor nodes. A similar approach exploiting roles of sensor nodes is proposed in [9].

Our middleware approach differs from those cited above by that we are focusing on a very specific functionality, i.e. security instead of trying to provide a communication or programming abstraction. In our approach flexibility is addressing support of a wide range of applications and individual support of the security needs of each application. I.e. we are trying to provide a tailor-made security solution for each application, and provide means to update the security modules during the runtime if necessary. In order to achieve this goal we are working towards a middleware compiler which selects security modules based on application and sensor node requirements and constraints respectively. In this area some work has been done, which did not focus on WSN and security issues but aims at a similar goal, i.e. providing tool support for development of a certain middleware. Most of those approaches are based on model driven architecture (MDA) [2]. The tool sets Cadena [7], VEST [14] and CoSMIC [1] are MDA based and try to support development of platforms for embedded systems. By that they provide functionality similar to our approach, the difference is that we are focusing on security and do not use MDA but defined our platform ar-

chitecture independently of any formal model. In addition only VEST supports the modelling of security aspects.

In [5] the authors discuss the integration of security aspects into a formal method based development of networked embedded systems. The focus of the security analysis language (SAL) is merely on information flow between networked entities. By that it might be a way to model security requirements of applications residing on top of our security modules and to verify whether or not our middleware compiler selected the correct modules.

3 Middleware Security Compiler

It is the goal to have a middleware that provides security functionality in a very flexible and adaptable way.

The security provided by our toolbox can be tailor-made on a per application basis and even be adapted during the life cycle of a certain application. This level of flexibility is achieved by a modular middleware architecture and by introducing the concept of a middleware compiler.

3.1 Overview of the Compiler Architecture

Since the gravity center of this work is security for WSN as such and not for a specialised application domain or even a single application, several solutions for each security service are required to be capable to provide security for a wide range of applications. This means that in order to provide flexibility of the choice there is a need for multiple modules that provide a specific functionality—service but differentiated with respect to security parameters, code size, etc.

The selection of the suitable security modules is done by our middleware compiler. In order to generate a suitable set of security modules for a certain application reasonable constraints need to be defined in advance. These constraints are on one hand due to the limitations of wireless sensor nodes and on the other hand imposed by the security that the application under development requires. The hardware driven constraints are for example processing power and available energy to name just a few. Application dependent constraints are lifetime of the overall network, security features like secrecy of measured data or similar. In order to define the relevant constraints an XML based description language for sensor nodes and application requirements is under development. Also the role of a specific sensor, if static or default, influences its software set-up. The sensor node description provides information concerning the hardware set-up of a sensor node plus relevant information of its software configuration such as operating system used and already allocated memory.

In addition to the description of the above mentioned constraints the security modules provide a self description. This description provides information concerning the functionality of the module, and the needed resources—memory footprint and processing power. If a module needs other additional modules its description also contains information on potential dependencies. For example,

an cipher mechanisms may require that a secure random number generator is also deployed.

Additionally, since the resources of a sensor node are usually constrained we suggest that every functionality that might be used by multiple modules is also present as a module in order to avoid code redundancy.

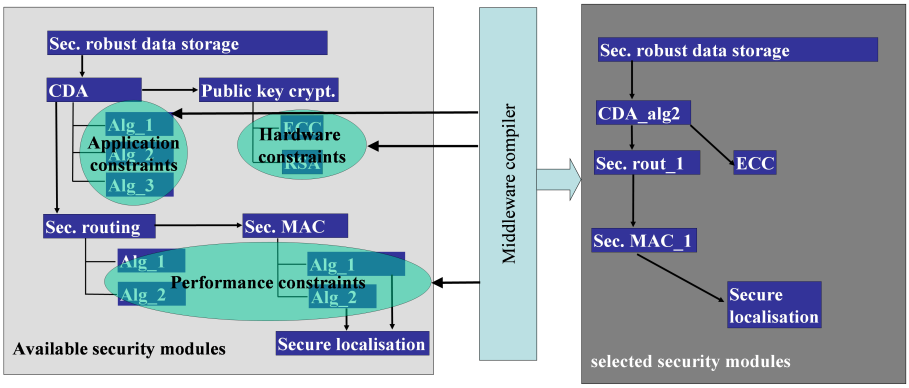


Fig. 1. The Middleware Compiler: the appropriate modules are selected based on application requirements and sensor node constraints.

Figure 1 illustrates the idea of the middleware compiler. The result of a successful compiler run is an instance of the secure middleware—the right hand side of the figure.

3.2 Compiler Operation

Our compiler approach requires the definition of abstract and concrete APIs between security modules. Using these interfaces the selection of the suitable security modules can be done by our middleware compiler. This compiler requires in addition the following information:

- required functions
- available modules
- constraints concerning performance and security

In following, these points are explained referring to an example. The example function is the authentication of another node.

Required functions It describes the needed functionality. In this example it is the authentication. The description does not contain any further specification details.

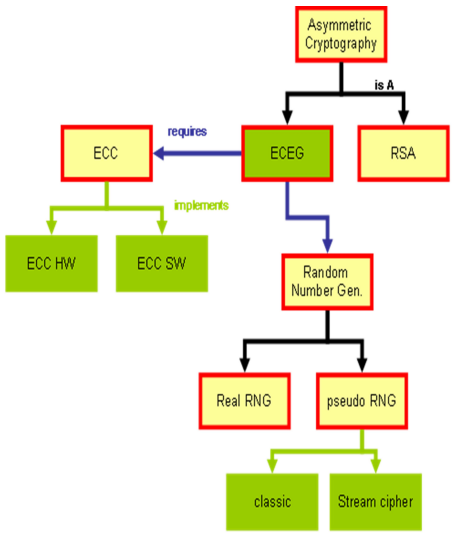


Fig. 2. Exemplary fragment of a dependency graph for asymmetric cipher mechanisms. Green boxes contain real code while bright boxes are logical classes—interfaces.

Available modules The description of available modules can be considered as a kind of database that contains every security module, its dependencies, interface description, security parameters, code size, etc. It is the notion that every module that is available has such a description. Based on these single module descriptions the compiler generates a network of modules, with dependencies and available implementations. Figure 2 shows an example of such a dependency graph. It is a part of the complete structure for our example, and as shown, there is a choice of the underlying mechanisms used by the authentication. It can be either ECEG (Elliptic Curve ElGamal) or RSA. If ECEG is chosen, an ECC (Elliptic Curve Cryptography) implementation and a Random Number Generator are required. For both of them diverse implementations are available.

Constraints concerning performance and security Based on the graph of available modules and the required functions the compiler determines possible configurations (i.e. sets of modules) that do not have open dependencies and provide every needed function. Indeed, some of these configurations will not meet the requirements of the system, either because hardware parameters are not met or due to wrong security properties. The description of such constraints helps to filter out configurations that do not fulfil the requirements. For instance, in our authentication example, there might be a constraint saying that the code size for the authentication functions must be less than 2 kB. In such a case, every configuration where the code size is more than 2 kB is withdrawn. Another possible constraint is security strength. If the example needs very strong authentication, all configurations based on weak cryptography implementations are withdrawn.

Final Evaluation At this point a set of possible configurations that meet every constraint, should be picked out. It is considerable that now the compiler evaluates extended parameters like energy consumption, total code size, performance, security implications and the like and chooses the best configuration or presents the results to the developer who chooses the best set-up.

4 Flexible Security Middleware Architecture

4.1 Overview

Starting on the highest level, there are two main classes of participants, i.e. sensor nodes and Configuration Centers. Figure 3 depicts our intended middleware architecture for both these classes. A Configuration Center runs applications that support the management of the WSN and allow access to data from specific sensors and aggregated data, and also to receive alarms. There may be multiple instances of Configuration Center, thus from now on the term Configuration Center refers to the class not to a single instance, if not otherwise stated.

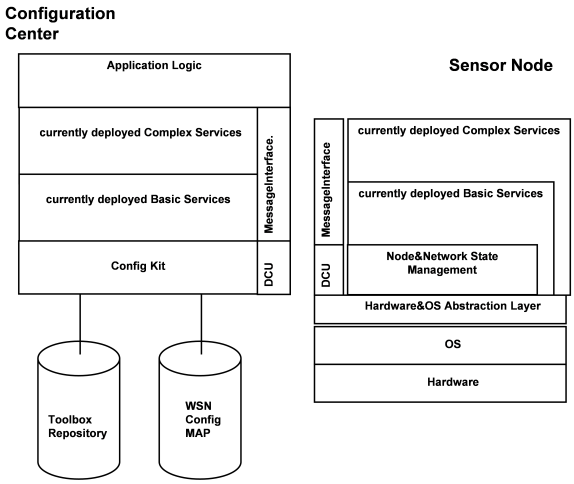


Fig. 3. Our middleware architecture: Configuration Center on the left and sensor node on the right hand side

Our middleware architecture distinguishes between three classes of components, where each component consist of one or more modules. These classes are:

- sensor node abstraction layer
- middleware core
- security services

The sensor node abstraction layer is the only operating system and hardware dependent component. It has to be adapted individually for each OS/hardware combination that shall be supported.

The middleware core consists of modules that are necessary to guarantee proper functionality of the other modules and those that are needed on all devices of a system based on our approach.

Security services are modules that contain the actual security functions, i.e. cryptographic means, security protocols and the like. The security services may use each other what implies a logic differentiation between basic services and complex services.

The general middleware architecture is mainly independent of the participant type, i.e. the deviations between sensor nodes and Configuration Centers are minor. The major differences concern the presence of the sensor node abstraction layer that is needed at sensor nodes, and the inclusion of the Middleware Compiler that is necessary only at the Configuration Center.

The concrete instantiation of the middleware, i.e. the modules deployed at the sensor nodes and at the Configuration Centers depends on:

- the currently running application
- the current role of the sensor node
- sensor node capabilities

4.2 Core Components

State Management Module (SMM) The SMM monitors the sensor node and maintains its state. By that it can trigger a code update for example if the sensor node reaches the management state, which might be caused by expiration of timers or by external triggers such as detection of malicious actions.

Each sensor node can be in one of the following four states (see Figure 4):

- M0 off-the-shelf state: the node is equipped with basic functions needed for the initialization process. The initial software configuration is put on the node. In order to protect keys, this can happen in a secure environment. After that the node is ready for deployment.
- M1 initialization: after the sensor nodes are deployed they are performing the network set-up, e.g. exploring their neighbourhood, setting up routing information, etc.
- M2 normal operation: the sensor node executes its application specific task.
- M3 management: If an unexpected behaviour is detected, e.g. caused by environment or by an attack, the node will enter the management state. The reason for the interruption is analysed and appropriate actions are initiated. The management state is also entered when an code update has to be executed. Whether a dynamic code update or re-start of the sensor node is necessary, depends on the trigger which initiated the transition from M2 to M3. If DCU has to be executed the Configuration Center shall verify if only the requesting node has to be updated or whether other nodes in the network also need to be re programmed (see Figure 5)

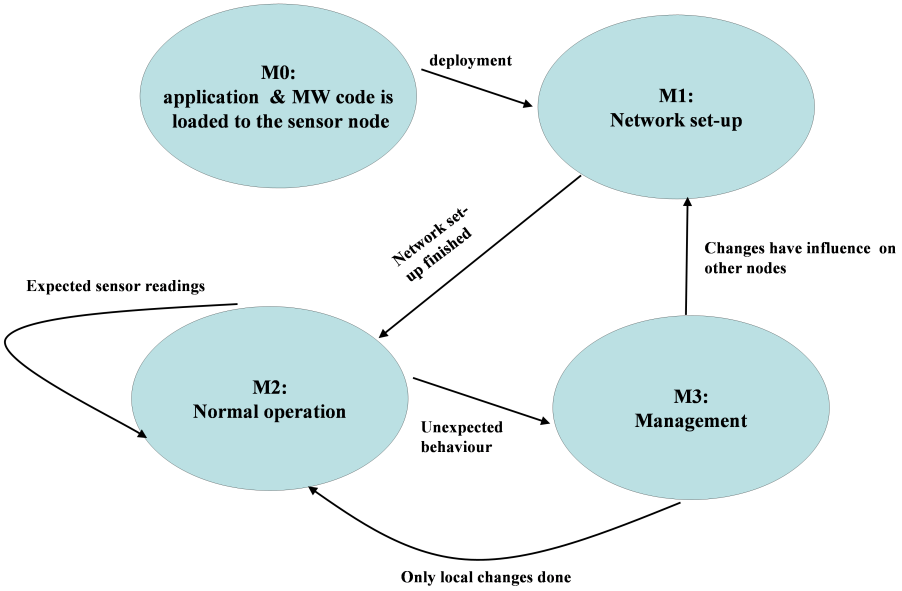


Fig. 4. Sensor node states before and after deployment

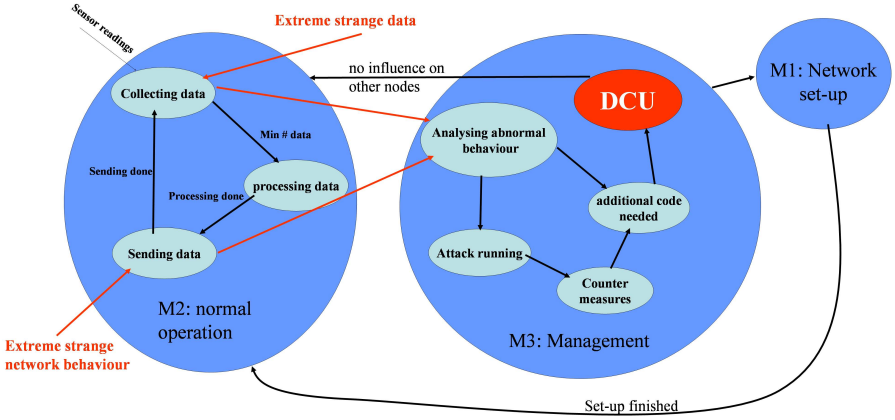


Fig. 5. State M2 and M3 including their internal states and events triggering the transition from M2 to M3 and vice versa

Message Interpreter The Message Interpreter provides local intelligence which is needed to decide for example if the current sensor node is capable to answer a query correctly or whether it has to forward the message. In addition it is a kind of middleware scheduler which passes incoming data to the corresponding middleware modules.

The message interpreter consists of two parts. It has a static part that is responsible for dealing with all messages that are directed to the middleware core components DCU and SMM. Its configurable part depends on the services deployed on the node. In order to properly support the configurable part the message interpreter uses a kind of registry which is shared with and maintained by the DCU module. Each time a module is exchanged, deleted or additionally installed the DCU module updates the registry. Thus, the message interpreter always knows which modules are available. Depending on the currently available modules and the node's current role and the message address the message interpreter decides what to do. In principle it is a three stages decision chain.

1. If the node is not the intended recipient the message forwarded.
2. If the node is the intended recipient the message interpreter checks if the corresponding module is deployed. If *yes* It is checked if the sensor node run in the appropriate role. If *yes* the message is delivered, otherwise it it forwarded to a more appropriate node.
3. If the node is the intended recipient but the corresponding module is not deployed the SMM is informed about this misalignment. The SMM can then decide what to do. Options are sending a misalignment message back to the configuration center, requiring a code update or just ignore the misalignment. The reaction of the SMM will depend on the sender of the message, i.e. if the sender is a well known trustworthy party some action will be taken otherwise the misalignment might be ignored.

Abstraction layer The abstraction layer provides generic interfaces to basic and complex services so they can be developed independent of the underlying operating system. Due to the nature of the security modules under development we foresee two interfaces: a storage and a communication interface. The former will provide memory management functionality such as allocation of memory, store and fetch operations of data items used by higher layers. The communication interface handles incoming and outgoing messages. The latter are passed as payload to the appropriate OS dependent interface. Incoming messages are passed to the message interpreter after removing all protocol headers and trailers if necessary, i.e. if the OS did not already remove them.

DCU This module is necessary to allow reconfiguration of sensor nodes during their lifetime. Potential triggers can be newly detected vulnerabilities of security modules or a simple reconfiguration due to deployment of new applications.

It can be assumed that there is a need for change or adaption of the security mechanisms also after the deployment of the nodes. It can be caused by a

changed network structure or size, or even by a broken security algorithm. This is why dynamic code updates are an substantial need of our security middleware. The diagram shown in figure 5 refers to DCU in state M3. That means that the system requires the capability to change the functionality running on the sensor node during runtime whenever it is needed by the management. In kernel based operating systems like Contiki[4] such dynamic updates are not very challenging. Processes can be added or stopped and executable code can be stored or removed from the node. In very resource efficient operating systems like TinyOS[8] that merge operating system and application to one image it is not that straightforward to change the functionality. Recently several mechanisms have been developed that provide code update functionality for TinyOS. Currently we are focusing on FlexCup [11],[12] that allows to change methods at runtime.

If the configuration of sensor nodes is changed during their life time, this is recorded at the WSN configuration map repository (see fig. 3). The repository always reflects the middleware instantiation of all sensor nodes starting from the first set-up. The current set-up of all nodes within a certain part or with a common task is used as an additional constraint whenever a code update is required after deployment. By that interoperability inside the WSN can be guaranteed, e.g. the use of different aggregator node election algorithms can be avoided.

4.3 Service Layer

As already mentioned there are two logic classes of services. This has been reflected in Figure 3 as well. The services are build from modules. In general, one module can use other modules, i.e. can use the functionality provided by other modules. This dependency causes that some of the services are on the top of the stack or tree, what actually causes that their functionality is more complex compared to services from lower parts of this structure.

This differentiation here is completely independent from the kind of functionality provided. It is only based on the information if a service is used by another or not.

Basic services are modules that do not support several functionality but just one. But they may rely on other basic services such as cipher means do since they require random number generators to be deployed. An example of basic service may be the secure random number generator presented in [3].

Complex services have more complex functionality, e.g. aggregation and persistent storage of sensor readings can be provided by the tinyPEDs service [6]. In order to fulfil their tasks such services may require support from basic services, e.g. to do encryption or decryption. But a complex service may also be implemented in a monolithic way so that it does not need any basic services to fulfil its tasks. Therefore the complex services may access the abstraction layer directly as basic services do.

5 Conclusions

In this paper we have introduced a flexible security providing middleware approach. Our way to achieve flexibility is twofold. On one hand we propose the use of a configuration tool that compiles a security architecture at development time, and on the other hand we provide an architecture that allows for dynamic exchange of security modules at run time.

The use of a middleware compiler like approach ensures that the application programmer no longer needs to be also a security expert, and by that reduces the probability of mis-configurations. The capability of up-dating the security configuration during run time is especially beneficial for long-living applications. It allows to exchange successfully attacked security means against still unbroken ones. The re-compilation of the security configuration of a certain sensor node is also executed by our configKIT.

We are currently finalising the XML based description languages for security modules, sensor node description and security requirements of the application. Our next research steps concern the selection of security modules and means to prove that the combination of the selected modules really provides the required security means at the required security level.

Acknowledgments

The work described in this paper is based on the results of IST FP6 STREP UbiSec&Sens. UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. Krishnakumar Balasubramanian Arvind. Applying model-driven development to distributed real-time and embedded avionics systems. *International Journal of Embedded Systems*, Special issue on Design and Verification of Real-Time Embedded Software, April 2005.
2. A. Brown, J. Conallen, and D. Tropeano. *Models, Modeling, and Model-Driven Architecture (MDA)*, chapter Introduction:. Springer Verlag, 2005.
3. C. Castelluccia and A. Francillon. Tinyrng, a cryptographic random number generator for wireless sensor network nodes. In *5th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, IEEE WiOpt*, 2007.
4. A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *In First IEEE Workshop on Embedded Networked Sensors*, 2004.
5. Matthew Eby, Jan Werner, Gabor Karsai, and Akos Ledeczi. Integrating security modeling into embedded system design. In *International Conference and Workshop on the Engineering of Computer Based Systems*. IEEE, 2007.

6. J. Girao, D. Westhoff, E. Mykletun, and T. Araki. Tinypeds: Tiny persistent encrypted data storage in asynchronous wireless sensor networks. *Ad Hoc Networks Journal (Elsevier).*, to appear.
7. J. Hatcliff, W. Deng, M. Dwyer, G. Jung, and V. Prasad. Cadena: An integrated development, analysis, and verification environment for component-based systems. In *Proceedings of the 25th International Conference on Software Engineering*, 2003.
8. J. Hill, P. Levis, S. Madden, A. Woo, J. Polastre, C. Whitehouse, R. Szewczyk, C. Sharp, D. Gay, M. Welsh, D. Culler, and E. Brewer. TinyOS: http://www.tinyos.net, December 2005.
9. Manish Kochhal, Loren Schwiebert, and Sandeep Gupta. Role-based hierarchical self organization for wireless ad hoc sensor networks. In *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 98–107, New York, NY, USA, 2003. ACM Press.
10. Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. TinyDB: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.
11. Pedro José Marrón, Matthias Gauger, Andreas Lachenmann, Daniel Minder, Olga Saukh, and Kurt Rothermel. *FlexCup: A Flexible and Efficient Code Update Mechanism for Sensor Networks*. In *Wireless Sensor Networks: Third European Workshop, EWSN 2006*. Springer-Verlag, 2006.
12. A. Poschmann, D. Westhoff, and A. Weimerskirch. Dynamic code update for the efficient usage of security components in wsns. In *Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN)*, 2007.
13. Kay Römer and Friedemann Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.
14. J. Stankovic, R. Zhu, R. Poornalingam, C. Lu, Z. Yu, M. Humphrey, and B. Ellis. Vest: An aspect-based composition tool for real-time systems. In *Proceedings of the IEEE Real-time Applications Symposium*, 2003.
15. Jan Steffan, Ludger Fiege, Mariano Cilia, and Alejandro Buchmann. Towards multi-purpose wireless sensor networks. In *ICW '05: Proceedings of the 2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05)*, pages 336–341, Washington, DC, USA, 2005. IEEE Computer Society.
16. Website. Ubiquitous sensing and security in the european homeland *http://www.ist-ubisecsens.org/*.
17. Kamin Whitehouse, Cory Sharp, Eric Brewer, and David Culler. Hood: a neighborhood abstraction for sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 99–110, New York, NY, USA, 2004. ACM Press.
18. Yong Yao and J. E. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, 31(2):9–18, September 2002.
19. Stephen S. Yau, Fariaz Karim, Yu Wang, Bin Wang, and Sandeep K. S. Gupta. Reconfigurable context-sensitive middleware for pervasive computing. *IEEE Pervasive Computing*, 1(3):33–40, 2002.

Context distribution using context aware flooding

Koen Victor, Julien Pauty, Yolande Berbers

KULeuven
Department of Computer Science

Abstract. Due to the proliferation of small networked mobile devices, the number of (indirectly) interconnected services in Ambient Intelligence (AmI) environments may grow without bound. The network contains a potentially enormous amount of context aware services that sense, gather and distribute context information. Without a central context repository, or a central server that locates the context information, it is a challenge to address parts of the environment that contain relevant context information.
In this paper, we propose a model and algorithm for context gathering and distribution that imposes a virtual structure on the network that aligns with the actual context information within the network. Distribution of context uses an adapted form of flooding, that is context aware. Context aware flooding allows to specify the environment in which context is relevant.

1 Introduction

In an Ambient Intelligence (AmI) environment, numerous services make context information available to the environment. This context information is used by services to enhance their operation by adapting their behavior. Context information may guide service adaptation and service composition, help service configuration or trigger service events.

Context gathering and distribution in an AmI environment is a difficult task. The network is a combination of structured and unstructured networks that have both fixed and mobile services. Without an overlay network, there is no global addressing scheme to access services or their content. Additionally, services may enter or leave the network without notice and do not have any knowledge of the environment in advance.

Structuring context information in an AmI environment typically involves using a central server for context gathering and storing, while using a context middleware system on each device in the network. However, in some environments, devices have limited capabilities, pe. a kitchen with a coffee machine, an oven, a fridge, lightening controls and a washing machine. These systems cannot run a large middleware system. In other environments like a corridor, a meeting room,... services are extremely volatile because they run on mobile devices. In such situations it is not feasible to choose a coordinating central server.

To tackle these problems, we structure the network using the context information itself, and not use the addressing scheme of the underlying network. To do this, we rely on the type of the context data that is distributed in the network. Context data is encapsulated in a Context Item (section 3) that adheres to a type system. The type system is common for all services. To distribute or retrieve a Context Item, we use flooding, but only nodes for which the Context Item is relevant store and distribute it further. To determine the relevance of a Context Item, it is associated with Distribution Requirements (section 4.2) that describe its destiny.

If the context information arrives in a part of the network where it is not relevant following the Distribution Requirements (section 4.2), the distribution stops. We also allow the relevance of a context type to decrease gradually, to make context data travel through parts of the network where the Distribution Requirements are not fulfilled.

In the following sections, we first discuss the representation of context and context types. Then, we go into the different context events that are used in the system. After that, we discuss the context aware flooding that propagates the context events through the network.

2 Related Work

Cohen et al. [1] focus on the architecture for a context distribution system. The authors use the producer-consumer approach to disseminate information. The representation of Context Items is similar to our approach. A simulation is performed to compare different distribution algorithms, however none of them takes the approach of addressing the relevant part of the network using Context Items. Sygkouna et al. [2] study efficient decentralized usage-aware search mechanisms. The approach to making context distribution and searching more efficient in this paper is using context usage patterns to restructure the context in the network. This approach is complementary to our context-aware flooding. Henricksen et al. [3] compare five middleware systems for context distribution and introduces the middleware PACE. The paper explains that Context Fusion Networks may be used for the aggregation of context and support a high degree of device mobility. Mobility is supported by buffering during times of disconnection. As we will explain in the paper, we buffer some Context Events to deal with an inconsistent context state which may be due to disconnection of services.

T. Gu et al. [4] describe the use of multiple overlays to cluster peers in a P2P-network based on pre-defined ontologies to provide an efficient decentralized search engine for context data.

3 Representation of Context Items

We base the representation of context on the key-value model [5] because of its simplicity and compactness. Although other models may be more expressive (ontology, object oriented, ...), there is no added benefit for the illustration of

context aware flooding. However, context aware flooding may be adapted to be used in conjunction with other models. The template for a Context Data item is as follows: `Context Type, Type Version, [(Key, ValueType)]`

The Context Type refers to a type in a general type system that is used by all services. For each Context Type, the key list of (Key, Value) pairs is predefined. Values can be integers (`integer`), a string (`"string"`), a range (`integer-integer`) or an enumeration of possible string values (`string | ...`).

`Type Version` allows to differentiate between different versions of the context types. Services only have to know the types that are relevant for their execution. The most important context types describe situations that are relevant for the user. The way the instantiated context types are sent out, is described in section 4.

3.1 Examples

In this section we present some of the context types we used to study context aware flooding. An extensive list of the used types is out of the scope of this paper. The following is an exemplary context template for a temperature sensor and a possible instantiation.

Temperature, 1, (Value, integer), (Metric, (Celsius | Fahrenheit))

`Temperature, 1, 20, Celsius`

The following template is used by all services the user has eye contact with when interacting with it. For example, we consider a DVD player. If the user presses pause and does not use the player for a while, then it may notify the environment that the user has stopped watching the DVD disc or DVD player menu.

HumanWatch, 1, (Value, (Yes| No | Maybe)), (Trust, (High | Medium | Low))

`HumanWatch, 1, No, High`

If the user presses pause using the remote of the DVD player, the remote would instantiate a HumanTouch type, that signifies that the user touches, or holds, the remote.

HumanTouch, 1, (Value, (Touch | StopTouch)), (Trust, (High | Medium | Low))

`HumanTouch, 1, Touch, High`

After some timeout, the remote creates:

`HumanTouch, 1, StopTouch, Medium`

The Trust is only medium here, because the remote does not know for sure that the user stopped touching the remote. It only knows the user did not press any buttons for a while.

For services concerned with the meaning of a location (this room, this floor, ...), we introduce the following context type:

FixedLocationService, 1, (Location, (Room | Floor | Building | Exterior)), (LocationId, integer)

Using this context type, a service can notify other services that it has a fixed location and that it can be used as a reference point. This is further discussed in section 4.3.

4 Context Events

A Context Event is a message sent by a service to notify other services that its context has changed, or to request a context item. Context changes may occur because a sensor has measured a (new) value, or an incoming Context Event has changed the context of the service.

A Context Event consists of the Context Item (section 3) and meta data that informs the environment about the service and the relation of the service with the delivered data. The service associates Distribution Requirements with each Context Event to guide the distribution of the event in the environment. In the following sections, we explain the notation of context events.

4.1 Context Event template

The template of a Context Event consists of five entries:

Distribution Type specifies whether the Context Item in the Context Event is a request (*Pull*), new information for the services in the network (*Push*), or an answer to a request(*Ans*).

Context Item is discussed in section 3. If the distribution type of the Context Event is a *Push* or *Ans* event, the (Key,Value) Context Item pairs contain the information about the Context Item that has to be delivered. If the Distribution Type is *Pull*, the Value of a Key may specify to what values it should correspond, as explained in section 3.

ServiceID identifies a service in the network.

EventID identifies an event. When combined with the ServiceID, the event is uniquely identified within a network.

Distribution Requirements specifies in what environment the Context Item is relevant. The Distribution Requirements are expressed using the Context Item templates. If the Context Event arrives on a service in the network that has a set of Context Items that correspond to the Distribution Requirements, the Context Event is in a relevant part of the network.

Source Environment is optional. It is only used in Pull Context Events and has the same format of the Distribution Requirements. When a Pull request is answered, the Context Item templates in Source Environment become the Distribution Requirements of the Ans Context Type.

The template for a Context Event is *(Push | Pull | Ans), Context Item, EventID, ServiceID, Distribution Requirements*. The action taken by the context aware flooding algorithm depends on the Distribution Type, the context of the service and the Distribution Requirements. This is explained in section 5. In the next section, we discuss the *Distribution Requirements*.

4.2 Distribution Requirements

The Distribution Requirements specify to what services the Context Event is relevant, using Context Item templates. The meaning of the Distribution Requirements depends upon the Distribution Type of the Context Event. If the

Context Event is a *Pull* event, the Distribution Requirements specify the context of the services that may provide an answer to the request. If the Context Event is a *Push* event, the Distribution Requirements specify the context of the services for which the sent Context Item in the Context Event is relevant. The Distribution Requirements consist of the following entries:

Available Context Items with Constraints is a list of Context Items that the service that gets this context event should have to be relevant. Constraints on the instantiated context templates can be specified by using a Value range in the (Key, Value) pair.

Soft Hop Count is the number of services the information may travel in a relevant environment. This imposes a hard bound on the number of relevant services that receive this Context Event. This avoids unbounded flooding of Context Events that do not have a good specification of the target environment.

Hard Hop Count is the maximum number of services a Context Event may travel through services that are not in an environment that complies to the Distribution Requirements. Sending Context Events across parts of the network that are not relevant is necessary to target relevant parts of the network surrounded by irrelevant parts. This imposes a hard bound on the number of non relevant services that receive this Context Event.

The template for Distribution Requirements is
Soft Hop Count, integer, Hard Hop Count, integer, Available Context Items with Constraints, [Context Items]

4.3 Examples

As an example we present three Context Events of a user who watches a DVD movie. On startup, the DVD player sends a context event that it is fixed in a location, in room with ID 912 (table 1).

Template part	Value	Description
Distribution type	*Push*	
Context Data	FixedLocationService, 1, Room, 912	This service is in Room 912.
EventID	1	First event
ServiceID	3452X01ZP	Unique service ID: the DVD
Distribution Crit	100, 20, -	There are no Distribution Requirements except for the number of hops.

Table 1. Context Event: startup of DVD player

A similar event is sent by the lightening system and the audio system. When the user presses play on the remote, it sends the Context Events described in

table 2. In the next section we discuss the algorithm to distribute these context events.

Template part	Value	Description
Distribution type	*Push*	
Context Data	HumanTouch, 1, Touch, High	The user touches this device.
EventID	1	First event
ServiceID	3EZ29X01ZP	Unique service ID: the remote
Distribution Crit	20, 20(FixedLocationService, 1, Room, 912)	This event is to be distributed to services that associate with Room 912.
Distribution type	*Push*	
Context Data	HumanWatch, 1, No, High	The user watches something
EventID	2	Second event
ServiceID	3452X01ZP	Unique service ID: the DVD player
Distribution Crit	20, 20(FixedLocationService, 1, Room, 912)	This event is to be distributed to services that associate with Room 912.

Table 2. Context Event: user uses remote of DVD player

5 Context-aware Flooding

In this section, we present the algorithm to distribute events in the network. First we explain how the algorithm deals with possibly incomplete context information. We discuss the meaning of receiving and sending Context Events and refer to the flooding algorithm implementation.

5.1 Dealing with inconsistent context states

When a service receives a Context Event, it has to decide whether the enclosed Context Item is part of its context. If the Context Item is part of its context, it is put in the context container of the service. The decision is based on the Distribution Requirements of the Context Event and the context of the service. The services need to make this decision with incomplete information because there may be inconsistencies between the context of the service as described by the service's context container and the context of the service in the real world. There is a delay between the sending of a *Push* Context Event and the receiving by the relevant parts of the environment. As such, a (*Pull*) Context

Event requesting information may travel through a part of the network that has no up-to-date context information. In relation to this, the relevance of a Context Event is based on the Distribution Requirements that depend on other Context Items in the context of a service. Therefore a Context Event may be considered relevant when it is not, or not relevant when it is.

We say that services that have complete context information (there are no more Context Events traveling in the network that are relevant for the services) have reached a *consistent context state*. Services that have context information that is not up-to-date are in an *inconsistent context state*. The service is part of the environment described in the Distribution Requirements, but is not (yet) aware of it. For a service, there is no way to determine whether its context is in a consistent or inconsistent state. There are two possible causes of an inconsistent context:

1. A Context Event specifying the context of the service is not yet received by the service.
2. There is a dependency between Distribution Requirements (Figure 1) and at least one of the dependencies has not yet arrived. For example, the relevance of Context Item CI_A may depend on the distribution requirement DR_X, which expresses a constraint on a Context Item CI_B and CI_C. When an event about CI_B arrives, it may depend on DR_Y that expresses a constraint on CI_C and CI_D. For CI_A and CI_B to become relevant at this service, CI_C and CI_D need to be available so DR_Y and DR_X can become fulfilled.

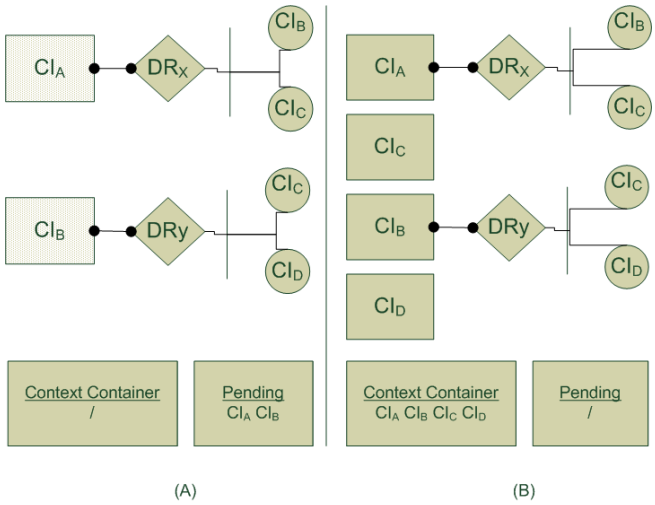


Fig. 1. (A) CI_A and CI_B are pending (B) CI_C and CI_D arrived in the context container

There are four different ways of dealing with services with an inconsistent context.

1. We may *ignore* the problem. In an environment wherein Context Events are sent frequently and in an arbitrary order, eventually all dependencies will be resolved.
2. A *central server* may resolve all dependencies and keep track of the Context Items in the context of a service. This is not an option for our decentralized AmI environment.
3. We may make an approximation of the consistent context state.
4. We may defer the decision whether an incoming Context Event is relevant or not, by caching Context Events that may become relevant in the future. This is the approach taken in this paper and is explained in the following section.

5.2 Caching Context Events

To deal with services in an inconsistent context state, we provide a cache for Context Events that are not relevant at the moment of arrival, but may become relevant in the future. We do not know whether a service is in an inconsistent context state. Therefore, we assume that a service may be in an inconsistent context state if a Context Event arrives that is considered irrelevant. We cache the Context Event until the service is in a consistent context state or the cached Context Event is deleted. Deletion of Context Events implies a management strategy for the Context Event. We do not cover the management strategy for context because an in-depth discussion context management is outside the scope of this paper. We consider the service in a consistent context state with respect to a Context Event, when the cached Context Event becomes relevant.

The following types of Context Events may become relevant only after the delivery of a service:

- *Push* events with distribution specifications that do not match the service's Context Container.
- *Pull* events with distribution specifications that do not match the service's Context Container.
- *Pull* events with distribution specifications that do match the service's Context Container but for which the context container does not provide an answer

Caching all these types of events implies that for every service a big part of the state of the network is held in memory. On mobile devices with limited capabilities this is not an appropriate solution. However, we observed that we may limit the number of cached Context Event types by only caching Push events. When a service issues a Context Event on the network, it wants its context information delivered to all services for which it is appropriate. However, there is never a guarantee that a Pull request is answered. The issuer of the Pull request can not know in advance that the requested Context Item is available

in the target environment. Therefore, a service that requests a Context Item may issue several Pull requests periodically, until the requested Context Item is received. As a result, Pull requests will arrive at their destination as the state of the networks evolves by the Push events, even when Pull requests are not cached.

Algorithm 1 update_environment

```
 1: parameters: ContextEvent ce, ContextItem ci, ContextType ct, Criteria crit
 2: complies = true;
 3: for each criterium c in crit do
 4:    for each Key k in c.getContextItem().getKeyValues() do
 5:       if (!k.complies()) then
 6:          complies=false; break;
 7:       end if
 8:    end for
 9: end for
10: if (complies & ce.getDistributionType() == Pull) then
11:    if (context.getContextEvent(ct) != null) then
12:       send(context.getContextEvent(ct), all);
13:    else
14:       send(ce, all);
15:    end if
16: end if
17: if (!complies & ce.getDistributionType() == (Pull or Ans) then
18:    send_hardhop(ce,all)
19: end if
20: if (complies & ce.getDistributionType() == Ans) then
21:    send(ce,all)
22: end if
23: if (!complies & ce.getDistributionType() == Push) then
24:    pending.add(ce)
25:    send_hardhop(ce,all)
26: end if
27: if (complies & ce.getDistributionType() == Push) then
28:    add(ce)
29:    if (requests.getContextEvent(ct) != null) then
30:       send(context.getContextEvent(ct),context.getContextEvent(ct).getSender())
31:    else
32:       send(ce, all)
33:    end if
34: end if
```

5.3 Receiving a Context Event

A Context Event may be received by every service that is not further away than specified in the hop count requirement of the Distribution Requirements. Upon

Algorithm 2 add

```
1: parameters: ContextEvent ce
2: context.add(ce)
3: check_criteria(ce.getCriteria())
4: update_environment(ce, ce.getContextItem(),
5: ce.getContextItem().getContextType(), ce.getCriteria())
```

receiving a Context Event, the service checks whether the event is relevant for the service. A Context Event is relevant for a service if the service complies to all the Distribution Requirements specified in the event. Also, the purpose of the distribution type is checked: *Push*, *Pull*, or *Ans*. In the following, we refer to the relevant line numbers of Algorithm 1 on page 9.

Receiving a Pull Event If the Context Event is not relevant *(l 17-19)*, the service sends the Context Events to the other services connected to this service, with the Hard Hop count decreased by one.

A *Pull* Context Event means that a Context Item is requested from the network. If the Context Event is relevant *(l 10-16)* and is contained in the context of the service, an *Ans* Context Event is sent to the sender of the Context Event containing the Context Item.

Receiving a Push Event A *Push* Context Event that is not relevant *(l 23-26)* may change the context of the service in the future, when the context of the service changes and the Distribution Requirements are fulfilled, as explained in section 5.1 . Therefore, we cache the Context Event.

A relevant *Push* Context Event *(l 27-34)* changes the context of the service: the Context Item contained in the event is added to the context of the service. Next, the Context Event is sent to all services connected to this service. The Soft Hop counter of the Context Event is decreased by one. Following, the algorithm checks whether there are Context Events that became relevant because of change of context. If this is the case, the cached Context Event is removed from the cache and is handled as a new incoming Context Event. As a result, the cache of other services may be updated too because the Context Event will be flooded again.

5.4 Sending a Context Event

A Context Event is sent by a service to distribute its Context Items or request Context Items from other services in the network.

A service that wants to distribute a Context Item, for example a temperature sensor, creates a *Push* Context Event. In the Distribution Requirements, it specifies the context characteristics of the target environment. For example, it may specify that the service should be in the same room as the sensor. Only services that are in this room will interpret this Context Event and store it in their

context. The Context Event will only travel outside this target environment for a limited number of services, as specified by Hard Hop Count.

A service that needs a Context Item issues a *Pull* Context Event. In the Distribution Requirements, it specifies the context characteristics of the environment where the service expects that the relevant Context Event is present. For example, it may request the temperature Context Item, and the Distribution Requirements may specify that the context item should be present in a particular room. The Context Event will only travel outside this target environment for a limited number of services, as specified by Hard Hop Count.

5.5 Sending and receiving an Ans Context Event

(l 17-22) When a Pull Context Event arrives at a service that has the requested Context Item in its Context, an Ans Context Event is issued and sent back over the connection from where the Pull request came.

The Context Items in the Source Environment of the Pull Context Event becomes the Distribution Requirements of the Ans Context Event. Upon receiving an Ans Context Event, the Context Item is added to the context of the service if it relevant following the distribution specifications. The Context Event is distributed using the same principles as sending another type of event.

5.6 Initialization of a service

Every context-aware service that connects to the network may cause services to have an inconsistent cache. Therefore, a service that contains a Context Item that is relevant for other services in the network, issues a Push event to make other services aware of its presence. This initialization Push event is not different from other regular Push events.

5.7 Implementation

The context-aware flooding algorithm is implemented as a reactive system that reacts on incoming Context Events. In the case that the service adds a Context Item to its own context, it creates a Context Event that is distributed to other services and itself. The key functionality of context-aware flooding is implemented in three algorithms.

- **check_criteria** checks each criterium of the Distribution Requirements of a Context Event. If the context of the service complies to a requirement, it is marked.
- **update_environment** is responsible for updating the context of the service and for sending messages.
- **add** is called from *update_environment*, and adds Context Events to the context of a service. The add method makes this algorithm recursive, because if the context of a service is modified, the *update_environment* has to check if other updates to the environment are required.

The *update_environment* and *add* algorithm are shown in Algorithm 1 and 2.

complies(Value, ValueCriterium) checks whether Value complies to the ValueCriterium. The values of the Context Items in the Context are compared to the Values. **send_hardhop(ContextEvent, destination)** sends to destination in the case that this service does not fulfill the Distribution Requirements. **send(ContextEvent, destination)** sends to destination in the case that this service does fulfill the Distribution Requirements.

6 Conclusion & Future Work

We have developed context aware flooding to gather and disseminate Context Items in an AmI environment. Distributing Context Items in AmI environments is a difficult task because of the heterogenous network setup and the volatility of services. We have explained how Context Items can be addressed using a specification of the part of the network where these Context Items may be found. We use Context Events to encapsulate the Context Items and the Distribution Requirements, that guide the Context Event through the network. To distribute Context Events, we use an adapted form of flooding, that is context aware.

We are working on a simulation in OMNET++ to evaluate this work. We expect that specifying Distribution Requirements will be more intuitive and flexible to the programmer of context aware programs than other ways of finding Context Items, like using a central server approach. As to performance, we expect a smaller number of messages in the network as compared to regular flooding.

References

1. Cohen, R., Raz, D.: An open and modular approach for a context distribution system. Network Operations and Management Symposium, 2004. NOMS 2004 (19-23 April 2004)
2. Sygkouna, I.; Anagnostou, M.S.E.: Efficient search mechanisms in a context distribution system. Advanced Information Networking and Applications, 2006. AINA 2006 (18-20 April 2006)
3. Henricksen, K., Indulska, J., McFadden, T., Balasubramaniam, S.: Middleware for distributed context-aware systems (2005)
4. T. Gu, E.T., Pung, H.K., Zhang, D.Q.: Contextpeers: Scalable peer-to-peer search for context information. 1st International Workshop on Innovations In Web Infrastructure (2005)
5. Mitchell: A survey of context-awareness. Internal Technical Report, Lancaster University, Computing Department (March 2002)

Coordinating Context-aware Applications in Mobile Ad Hoc Networks

Jean-Marie Jacquet[1] and Isabelle Linden[1]

Institute of Informatics, University of Namur
Email: {jmj,ili}@info.fundp.ac.be
WWW home pages: http://www.info.fundp.ac.be/~{jmj,ili}

Abstract. Mobile ad hoc networks are opportunistically formed by mobile hosts moving in time from locations to others and thereby coming accidentally within wireless range and thus temporally able to communicate. Designing rich applications in that context is challenging but meets today demands. In this paper, we demonstrate the interest of our Bach coordination language for that purpose. Building upon existing features, such as synchronization on the availability of information and temporal primitives, we propose extensions to code context-aware applications, notably by providing facilities to dynamically change data flows when hosts connect and disconnect, to filter information and hosts and to cope with changes in connection quality.

1 Introduction

In the aim of building interactive distributed systems, a clear separation between the *interactional* and the *computational* aspects of software components has been advocated by Gelernter and Carriero in [8]. Their claim has been supported by the design of a model, Linda ([4]), originally presented as a set of inter-agent communication primitives which may be added to almost any programming language. Besides process creation, this set includes primitives for adding, deleting, and testing the presence/absence of data in a shared dataspace.

A number of other models, now referred to as coordination models, have been proposed afterwards (see [21, 22] for a comprehensive survey of many of them). Building upon our coordination language Bach, we turn in this paper to applications developed in the context of mobile ad hoc networks. Such applications are widely recognized as challenging because of the dynamic network topology and of the inherent unpredictability of the environment. However, as mobile computing platforms are increasingly being used, there is a pertinent need for coordination models that can support the development of applications on such platforms.

A crucial property to that end is the possibility to handle context in a reactive manner. For instance, two hosts should be able to share information when they come within wireless range. Furthermore, as the topology of a mobile ad hoc network is changing very dynamically, a clear decoupling both in time and space between producers of information and consumers of information is highly desirable.

Some of these requirements are already met by our coordination language Bach. Indeed decoupling information producers and consumers is provided by the Linda framework and subsequently by our coordination model Bach, on which it is based. Moreover, the blackboard relations of Bach can be used to dynamically link dataspaces. Furthermore, the timed primitives it incorporates allows to cope with disconnections when remote communication primitives are performed. However, in order to fully tackle the development of context-aware applications for mobile ad hoc networks, Bach is extended in this paper by mechanisms that dynamically activate blackboard relations when hosts connect and disconnect, that filter information and hosts and that handle changes in connection quality.

In that aim, the rest of the paper is organized as follows. Section 2 introduces our basic coordination model Bach and explains three important extensions aiming at handling distribution, time and relations. Building upon these pieces of work, section 3 presents our coordination mechanisms for mobile ad hoc networks. Finally, section 4 draws our conclusions and compares our work with related work.

2 The coordination language Bach

Linda ([4]) has proposed to model the communication of concurrent processes by means of a shared tuple space on which concurrent processes act by adding tuples, by checking their presence and possibly by consuming them. The action of the concurrent processes is asynchronous in the sense that a producer process enriches the tuple space without the need for a consumer process to be ready to consume it. Conversely, a process consults a tuple regardless of the process which has created it on the tuple space. As a result, processes synchronize on the basis of the availability of information and not on the basis of a possible synchronous communication on a channel, as in traditional synchronous process algebras like CCS ([17]), CSP ([10]) and π-calculus ([18]). Obviously this pattern of communication is independent of programming languages and is actually to be embodied in any programming language through a dedicated library.

The Linda idea is enjoying a wide acceptance in many scientific communities ranging from economics to artificial intelligence (see e.g. [16]). As we shall argue later, we believe that it is a good basis for coding context-aware applications in mobile ad hoc networks.

The idea of synchronizing processes on the basis of the availability of information has also been adopted later by so-called concurrent constraint programming ([23]) for which the availability of information is generalized to the entailment of constraints from constraints already present in a shared constraint space. Inspired by Linda and by concurrent constraint programming, we have developed several extensions. We subsequently review those extensions adapted to build a coordination language for mobile ad hoc networks.

2.1 The core language

As a snapshot, our core language proposes to use a single shared data space, also called the blackboard. It is populated by tuples, each one consisting of a list of ascii parameters. For example, $\langle namur, 12, 18\rangle$ is a valid tuple which may represent today expected lowest and highest temperatures in the city Namur.

The blackboard is accessed through four primitives: (i) $tell(t)$ to put the tuple t on the blackboard, (ii) $ask(t)$ to check its presence on the blackboard, (iii) $nask(t)$ to check its absence, and (iv) $get(t)$ to remove an occurrence of it from the blackboard. Although the *tell* primitive always succeeds, the last three primitives suspend as long as the presence/absence of the token t is not met. Moreover, in order to allow values to be taken from the blackboard, some fields of the tuple t may be replaced by variables. For instance, assuming the presence of the above tuple $\langle namur, 12, 18\rangle$, the execution of the primitive $ask(\langle namur, ?min, ?max\rangle)$ instantiates the variables min and max to 12 and 18, respectively. Nevertheless, in order to force the communication to happen only through the blackboard and not via shared variables, tuples told by the *tell* primitives can never contain variables unbound to values.[1]

As we intend to focus on communication, we shall model concrete programming instructions as abstract atomic operations and consequently consider processes as computing the above four primitives, these abstract operations, and their combination through the following three classical operators: " ; ", " || " and " + " to respectively denote the sequential composition operator, the parallel composition operator and the non-deterministic choice operator.

In order to allow iterative behaviors, we introduce recursion variables as in classical process algebras (see for instance [7]) and define their behavior by equating them to agents composed of the above four primitives, procedure variables and the above three composition operators. For instance, the procedure variable X defined by $X = tell(t)\ ;\ X$ infinitively produces occurrences of the tuple t on the blackboard. Such variables are typically denoted by strings starting with an upper case letter (while other constructs will be written with strings starting with a lower case letter).

It is worth stressing that it is quite easy to add new processes in an existing system. Indeed, all one has to do is to provide code for these new processes and to execute them concurrently with the other processes. The main reason for this ease of extension is that processes are decoupled from each other and their communication and synchronization only rely on the availability of information on the blackboard. In that respect, it is worth noting that the characteristics of mobile ad hoc networks, and particularly limited bandwidth and frequent disconnections, favor a decoupled style of programming. For this reason, we believe that the Bach coordination language is well suited for programming applications on mobile ad hoc networks.

[1] Variables bound to values are allowed and, as expected, are actually replaced by their values. More precisely, variables typed without interrogation mark are understood to be replaced by their values whereas variables with an interrogation mark are understood as place holders requiring values.

2.2 Timing issues

Coding practical applications evidences the fact that data and requests rarely have an eternal life. For instance, a request for information on the web has to be satisfied in a reasonable amount of time. More crucial is even the request for an ambulance which, not only has to be answered eventually but within a critical period of time. The list could also be continued with software in the areas of air-traffic control, manufacturing plants and telecommunication switches, which are inherently reactive and, for which, interaction must occur in "real-time".

In our recent work ([12–15]), we have proposed different ways of introducing time in coordination languages. For that purpose, we have used the classical two-phase functioning approach to real-time systems (see eg [2, 5, 9]) and have proved that this approach was effective for modeling coordination in reactive systems.

In two words, our approach may be described as follows. In a first phase, elementary actions of statements are executed. They are assumed to be atomic in the sense that they take no time. Similarly, composition operators are assumed to be executed at no cost. In a second phase, when no actions can be reduced or when all the components encounter a special timed action, time progresses by one unit.

In that context, four families of timed coordination languages have been introduced in [12] by incorporating delays and time bounds to the primitives either in a relative way or in an absolute way. The expressiveness of these families has been studied in [12–15]. For this paper, we shall use the relative time families and consequently add a *delay* primitive and enrich the *tell*, *ask*, *nask*, *get* primitives with durations, indicated in subscripts and taken as integers. More precisely,

- $tell_n(t)$ is used to put the tuple t on the blackboard for n units of time,
- $ask_n(t)$ is used to check the presence of the tuple t on the blackboard under the constraint that if this is needed, the primitive suspends only for n units of time, after which it fails,
- $nask_n(t)$ and $get_n(t)$ are similar to $ask_n(t)$ but respectively check the absence of the tuple t on the blackboard and remove an occurrence of the tuple t on the blackboard,
- $delay(n)$ is used to delay the computation by n units of time.

2.3 Multiple blackboards and distribution

Extending the timed Bach language to multiple blackboards on a single host is easy to imagine. Assuming a mechanism to define globally unique identifiers (which can be done for instance by defining a standard naming blackboard on the host), one has basically to augment each of the four primitives by a new argument specifying the blackboard on which the operation has to be performed. Concretely, $tell(bbn, t)$ aims at putting the tuple t on the blackboard named bbn. Similarly, $ask(bbn, t)$ requests the presence of the tuple t on the blackboard named bbn. Assuming a global clock for the host, the timed primitives of section 2.2 extends similarly to multiple blackboards.

Handling mobile ad hoc networks requires however to cope with blackboards on multiple hosts. This is more delicate for several reasons.

Firstly, different hosts may have different clocks running at slightly different speeds. In order to model this reality, we shall rely on the assumption that timed primitives refer to the clock associated with the host on which the blackboard to which they refer is located. Accordingly, the same timed primitive executed on different blackboards located on different hosts may have slightly different timing behaviors.

Secondly, hosts should be uniquely identified. We shall assume such a global unique identification for hosts (which could for instance be achieved thanks to mac addresses). Given that blackboards are uniquely identified on a single host, we then have a way of identifying blackboards in the context of multiple hosts. Indeed, a blackboard is then identified by the identification of the host on which it is located and its unique identification on that host. Concretely, we shall assume a set of location names, identify a host with a location name, and extend the blackboard primitives in order to mention the host name. As a result, the *tell*, *ask*, *nask*, *get* primitives respectively rewrite as $tell_d(bbn, t)@l$, $ask_d(bbn, t)@l$, $nask_d(bbn, t)@l$ and $get_d(bbn, t)@l$, where d is a duration, t is the considered tuple, l is a location name, and *bbn* is the name of the blackboard on the host identified by l. Moreover, a default blackboard is assumed on every location. Accordingly the arguments *bbn* and l are dropped to indicate that the blackboard primitives actually refer to the default blackboard on the local host. In that way, the model developed in subsections 2.1 and 2.2 is a special case of the extension to multiple blackboards under discussion.

Thirdly, the access to a blackboard is different according as the process performing the considered primitive is being executed on the host on which the blackboard is located or not. Indeed, as detailed in [12], blackboards may be implemented as pieces of memory which can be directly accessed by processes running on that host. In contrast, they have to be accessed by processes on different hosts by means of special mechanisms like sockets, remote procedure calls or remote methods invocations. This has lead us to define in [6] the notion of *virtual blackboards* which essentially are pointers pointing to other blackboards. As argued in [11], these links are particular cases of a more general concept called blackboard relations. Stated in declarative terms, blackboard relations assert the presence or absence of tuples on blackboards from the presence or absence of (possibly other) tuples on blackboards. For instance, the relation

$$in(b_1, X) \longrightarrow in(b_2, X)$$

is a forward relation which asserts that any tuple X on blackboard b_1 should also be considered as present on blackboard b_2. In general, blackboard relations take the following form:

$$\begin{aligned} &in(b_1, t_1), \cdots, in(b_m, t_m), nin(b_{m+1}, t_{m+1}), \cdots, nin(b_n, t_n) \\ &\quad \longrightarrow\ in(b_{n+1}, t_{n+1}), \cdots, in(b_p, t_p), nin(b_{p+1}, t_{p+1}), \cdots, nin(b_q, t_q) \end{aligned}$$

Such a relation expresses that the presence of $t_1, \ldots, t_m$ on blackboards $b_1, \ldots, b_m$ and the absence of $t_{m+1}, \ldots, t_n$ on $b_{m+1}, \ldots, b_n$ implies the presence of

t_{n+1}, ..., t_p on blackboards b_{n+1}, ..., b_p and the absence of t_{p+1}, ..., t_q on b_{p+1}, ..., b_q.

Operationally, the above declarative reading may be done in two ways.

- *Forward reading.*
  - Whenever a new tuple t_i $(1 \leq i \leq m)$ is put on b_i then for any tuple of tuples $(t_1, \cdots, t_m)$ from the tuple of blackboards $(b_1, \cdots, b_m)$ and for any tuple t_j not in b_j $(m+1 \leq j \leq n)$, the corresponding tuples t_k $(n+1 \leq k \leq p)$ should be created on b_k and the corresponding tuples t_l $(p+1 \leq l \leq q)$ should be removed from b_l.
  - Whenever a new tuple t_i $(m+1 \leq i \leq n)$ is removed from b_i then for any tuple of tuples $(t_1, \cdots, t_m)$ from the tuple of blackboards $(b_1, \cdots, b_m)$ and any tuples t_j not in b_j $(m+1 \leq j \leq n, j \neq i)$, the corresponding tuples t_k $(n+1 \leq k \leq p)$ should be created on b_k and the corresponding tuples t_l $(p+1 \leq l \leq q)$ should be removed from b_l.
- *Backward reading.*
  - The presence of t_k $(n+1 \leq k \leq p)$ can be deduced from the presence of a tuple of tuples $(t_1, \cdots, t_m)$ on the tuple of blackboards $(b_1, \cdots, b_m)$ and the absence of tuples $(t_{m+1}, \cdots, t_n)$ on $(b_{m+1}, \cdots, b_n)$.
  - The absence of t_l $(p+1 \leq l \leq q)$ can be deduced from the presence of a tuple of objects $(t_1, \cdots, t_m)$ on the tuple of blackboards $(b_1, \cdots, b_m)$ and the absence of objects $(t_{m+1}, \cdots, t_n)$ on $(b_{m+1}, \cdots, b_n)$.

Variants arise by considering the following issues:

- some tuples t_i $(1 \leq i \leq m)$ are consumed in the process while others are not;
- similarly, the evaluation of some $nin(b_j, t_j)$ $(m+1 \leq j \leq n)$ leads to the actual creation of tuples t_j on b_j while others do not;
- some tuples t_k $(n+1 \leq k \leq p)$ are actually created while others are not;
- tuples on b_l matching the tuples t_l $(p+1 \leq l \leq q)$ are removed as a result of the evaluation of some $nin(b_l, t_l)$ primitives while others do not have this effect.

For the ease of writing, we shall assume as a basic rule that the evaluation of the *in* and *nin* primitives induce modifications on the corresponding blackboards. Brackets are subsequently used to overcome this rule. For instance, $in(b, t)$ in the left-hand side of a rule states that t should be consumed on b while $[in(b, t)]$ states that it should not.

Very often, rules are written with only one of the forward or backward readings in mind. In these cases, these readings can actually be regarded as eager and lazy evaluations, respectively. To meet this intuition and to force the rules to be used in one of the modes only, we shall subsequently replace the $\longrightarrow$ arrow by the $\longrightarrow_f$ and $\longrightarrow_b$ arrows to respectively restrict the considered rule to its forward and backward readings.

Finally, interesting relations may be obtained by combining these rules in a natural conjoined form:

$$Rel \equiv \{R_1, \cdots, R_m\}$$

There Rel is a general relation defined as the conjunction of the elementary R_i rule relations. For instance, the above forward relation can be refined as follows:

$$forward(b_1, b_2) \equiv \left\{ \begin{array}{c} in(b_1, X) \longrightarrow_f in(b_2, X) \\ [in(b_2, X)] \longrightarrow_b [in(b_1, X)] \end{array} \right\} \quad (1)$$

It states, on the one hand, that any tuple put on b_1 should be removed and put on b_2 and, on the other hand, that any request for a tuple on b_1 should lead to a request for that tuple on b_2.

Note that in the above general relation, X is used to denote a variable in a logical way. As might easily be guessed, patterns of tuples may be defined by instantiating some fields and leaving others as logical variables. For instance

$$in(b_1, \langle namur, 12, X \rangle) \longrightarrow_f in(b_2, \langle namur, 12, X \rangle)$$

indicate that any tuple of the form $\langle namur, 12, \_ \rangle$ with any value for the third field should be forwarded to b_2.

As for tuples, relations can be told, asked and got. To that end, we introduce the primitives $tell(rn)$, $ask(rn)$, $nask(rn)$ and $get(rn)$. To continue the analogy with tuples, an interesting question is to ask where these relations are stored. Actually, it turns out that blackboard relations can be simulated by using the blackboard primitives acting on tuples as well as a few auxiliary operations. More specifically, in a stable network, relations between blackboards are stored on special blackboards on which suitable processes run. Accordingly, a call to a relation instance, say named rn, defined by the definition

$$Rel \equiv \{R_1, \cdots, R_m\}$$

is modeled by the creation of a blackboard named after rn, say $bbrn$, and on which a background process runs aiming at handling the inferences induced by the r-rules R_1 to R_n. We refer the reader interested by more details to [11].

As mobile ad hoc networks pose the problem of the unstability of connections and of the lack of centralization, alternatives will be proposed in the next section.

3 Bach as a programming model for context-aware mobile applications

Mobile ad hoc networks are opportunistically formed by mobile hosts moving in time from locations to others and thereby coming accidentally within wireless range and thus temporally able to communicate with each other. As a result, the topology of these networks is highly dynamic and unpredictable. Naturally, a key issue for developing applications on these networks is, for an host, to use the information offered by hosts in the vicinity. However, as experienced with the web, not all the information need to be propagated everywhere.

As described in section 2, the Bach model promotes the synchronization of processes on the basis of the availability of information and offers a complete

decoupling in time and space between the processes. It thus forms a good basis for developing applications in a framework such as mobile ad hoc networks where the parties involved change dynamically and for which context information is a key issue. However, the Bach model needs to be extended in three main directions.

Firstly, blackboard relations allow in principle to dynamically link blackboards and thereby render the information of the blackboard of a host available to processes running on a neighbourhood host. For instance, using the forward relation of equation (1) the information available on b_1 becomes available for processes only knowing b_2. However, as hosts may move out of connection and may reappear, a means has to be provided to dynamically activate and disactivate blackboard relations. This will be achieved by integrating events in the model and by making blackboard relations reactive to events.

Secondly, it may be desirable to filter information as well as hosts. Indeed, not all the information need to be accessible to any close host. Moreover, even if the information is available, not all hosts may perform all the actions. For instance, one may imagine that only specific hosts are allowed to remove tuples on a foreign host blackboard whereas others are just allowed to consult these tuples. To that end, an access policy is published when hosts start engaging a connection.

Finally, as blackboard relations allow a query to be satisfied by consulting different blackboards, a strategy has to be defined to determine which blackboard is to be consulted first. This is achieved by associating priorities with blackboard rules, which may vary in time in order to reflect changes in bandwidth.

Before describing these extensions, we first put a few restrictions on the Bach model developed in section 2, in particular on blackboard relations, so as to get a model viable in the context of mobile ad hoc networks.

3.1 Restrictions on Bach

As noticed at the end of section 2, the centralized implementation of blackboard relations cannot be transposed directly to mobile ad hoc networks. To that end, we require blackboard relations to be expressed in a forward or backward fashion only and furthermore constraint them to the following forms:

$$in(b_1,t_1),\cdots,in(b_1,t_m),nin(b_1,t_{m+1}),\cdots,nin(b_1,t_n) \longrightarrow_f in(b_{n+1},t_{n+1}),\cdots,in(b_p,t_p),nin(b_{p+1},t_{p+1}),\cdots,nin(b_q,t_q) \quad (2)$$

$$in(b_1,t_1),\cdots,in(b_1,t_m),nin(b_1,t_{m+1}),\cdots,nin(b_1,t_n) \longrightarrow_b in(b_2,t_{n+1}) \quad (3)$$

$$in(b_1,t_1),\cdots,in(b_m,t_m),nin(b_{m+1},t_{m+1}),\cdots,nin(b_n,t_n) \longrightarrow_b nin(b_2,t_{n+1}) \quad (4)$$

with the possibility of annotating some of the *in* or *nin* statements with brackets in order to avoid the consumption of data. As easily observed by the reader, the syntactic restriction allows the blackboard rules to be implemented as processes on one blackboard.

Moreover, time associated with the *tell*, *ask*, *nask* and *get* primitives provide a natural way of defining timeouts and of preventing from sudden disconnections. To that end, all the primitives are actually decoupled in two atomic actions: the starting of the action and the collection of the result. Consider the $ask_d(t)$ primitive executed by a process of the host l. It may be satisfied directly by the presence of the tuple t on the blackboard of the host l. In that case, the success (together with possible values for variables) is directly reported to the process executing the primitive. Otherwise, a backward blackboard relation may be used, which requires to search a distant blackboard. In that case, auxiliary primitives with duration d are used to search according to the relation. If all of them report success and if this success happens within d units of time according to the clock of l then a success is reported. Otherwise, failure is reported.

The execution of the $nask_d(t)$ and $get_d(t)$ primitives are similar. For the latter however, it may happen that the success of the removal of a tuple on a distant blackboard happens too late to consider the execution of the get primitive as successful. In that case the removed distant tuple should actually be placed back. This is achieved by annotating the distant tuple with the d duration and by confirming its deletion or by canceling it by corresponding messages. By default, the removal of the distant tuple is assumed to be canceled if no message reaches it before d units of time.

Accessing a distant blackboard *bbn* by primitives of the form $tell(bbn, t)$, $ask(bbn, t)$, $get(bbn, t)$ and $nask(bbn, t)$ is achieved through a special local blackboard *bbv* associated with the following blackboard relations:

$$\left\{ \begin{array}{c} in(bbv, X) \longrightarrow_f in(bbn, X) \\ [in(bbn, X)] \longrightarrow_b [in(bbv, X)] \end{array} \right\}$$

As any tuple told on *bbv* is forwarded to the distant blackboard *bbn*, it may result from connectivity problem that the tuple actually never reaches *bbn*. In that case, the primitive is considered as failed. This is handled by requiring the receipt of a confirmation message during the d units of time associated with the *tell* primitive.

3.2 Priorities

The description of the previous subsection leaves room for multiple blackboard relations executable on the insertion of a tuple or on the search for presence/absence of a tuple. To order the selection of blackboard relations, rules may be associated with an integer varying from 0 to 255. By default, rules without such a number specified are assumed to be associated with 255.

Rules are then employed with respect to the numbers with a higher priority given to rules associated with a higher integer. In case of equality, rules are taken in an arbitrary order.

Note that by getting rules and telling them back, it is easy to modify the priorities associated with the rules.

3.3 Events and reactive blackboard rules

In the context of mobile ad hoc networks, the ability to react to events and do so as soon as possible is of great importance. Examples of such events are the connections to new hosts, disconnections or changes in the quality of connection.

The Bach model provides a comfortable way of capturing events and of reacting on their presence. Indeed, events are naturally represented as tuples and reactions can be coded by blackboard rules (for instance, by forward rules).

Events are assumed to be generated by dedicated processes continuously monitoring the underlying layers of the Bach implementation. They are represented as tuples but, as we shall see in the next subsection, with special attributes forbidding ordinary processes to remove them. Three events are used in the Bach model: $\langle connect, h\rangle$, $\langle disconnect, h\rangle$, and $\langle quality, h, q\rangle$ to respectively denote the connection with host h, the disconnection with it and the quality of the connection.

On the other hand, although blackboard rules have the right shape, the reaction to an event requires generally an action and not the declarative assertion of the presence or absence of tuples. For instance, it is reasonable that upon connection, a host offers a forward relation from its blackboards to the blackboards of the connected host. To make the model simple, between hosts, we only allow the default blackboards of the hosts to be related. This does not introduce a lack of generality since, if need be, local blackboards may be linked with the default blackboard by means of blackboard relations. Thanks to this convention, the expected forward relation is obtained as follows:

$$in(\langle connected, X\rangle) \Rightarrow tell([in(Y)@self] \longrightarrow_b [in(Y)@X])@X \qquad (5)$$

where the $\Rightarrow$ arrow indicates that the forward rule is activated as soon as the tuple on its left hand side is present, where $self$ refers to the name of the considered host and where X denotes the host which has been connected to the considered host. As for blackboard relations, it is possible to enrich the left-hand side with other tuples (present or absent). Moreover, the right-hand side in its general form allows a sequence of arbitrary actions.

3.4 Access control

In order to prevent undesirable actions to occur on tuples and on blackboards, capabilities are associated with each tuple and with each blackboard primitive. Concretely, these capabilities take the form of hidden attributes of tuples, blackboard relations and primitives. They are inherited from the process which has lead to the creation of the considered tuple, the considered blackboard relation and to the execution of the considered primitive. An action on a tuple or a blackboard relation is only allowed if the capabilities are matched. For instance, a tuple reporting a connection is created by a system process and thus can only be removed by a process having the capability of the system process. Similarly, the forward relation (5) has the capability associated with the host under consideration and may be accepted or refused by host X according to its local policy.

4 Conclusion

The paper has proposed extensions to the Bach coordination model to tackle context aware applications in mobile ad hoc networks.

Before presenting them, we have argued that Bach main features form an attractive basis for the target applications. Indeed, Bach promotes a form of synchronization of processes based on the availability of information and, thereby, a physical and temporal decoupling of processes. These are crucial properties for coding applications in a dynamically changing network topology. Moreover, Bach blackboard relations are useful to dynamically allow hosts to share information when they come in wireless range. The interest of timed primitives has also been shown for handling possible disconnections when blackboard operations are performed.

The extensions we have proposed are threefold: (i) events and reactions to them in order to dynamically handle connections and disconnections, (ii) priorities to cope with the quality of the connections and (iii) capabilities to control actions on blackboards and to allow hosts to be filtered.

In the coordination community, Lime ([19]) is certainly the piece of work closest to ours. However there is in Lime no provision for timed primitives, blackboard relations, access control and priorities, which we found crucial. Indeed, as mentioned earlier, time provides good abstractions for coping with sudden disconnections. Moreover, blackboard relations allow to create dynamic links when connections and disconnections occur. As illustrated by the forward relation, they are able to express the transient sharing of tuple spaces offered by Lime but also allow finer control like expressing the absence of information. This last property also differentiates reactive rules of Lime and Bach: those of Lime only react to the presence of one tuple whereas our rules react on the presence and absence of a series of tuples. Finally, access control and priorities have been introduced to filter information and hosts as well as to cope with changes in quality of connections.

Other pieces of related work include TuCSoN ([20]) and MARS ([3]). Both provide programmable tuple spaces similar to our reactive rules. However, in contrast to our work, these reactions are designed to be implemented by manager agents only. Furthermore, in contrast to Bach and what is provided by mobile ad hoc networks, remote operations assume the persistence of connectivity.

Outside the coordination community, Mobile CORBA ([1]) embodies mobility for clients accessing distributed objects. However, in contrast to Bach, it is designed to work in nomadic computing environments where mobile clients rely on a stable networking infrastructure.

References

1. S. Adwankar. Mobile CORBA. In *DOA*, pages 52–63, 2001.
2. G. Berry and G. Gonthier. The Esterel Synchronous Programming Language: Design, Semantics, Implementation. *Science of Computer Programming*, 19, 1992.

3. G. Cabri, L. Leonardi, and F. Zambonelli. Mars: A Programmable Coordination Architecture for mobile Agents. *IEEE Internet Computing*, 4(4):26–35, 2000.
4. N. Carriero and D. Gelernter. Linda in Context. *Communications of the ACM*, 32(4):444–458, 1989.
5. P. Caspi, N. Halbwachs, P. Pilaud, and J. Plaice. Lustre: a Declarative Language for Programming Synchronous Systems. In *Proc. POPL'87*. ACM Press, 1987.
6. K. de Bosschere and J.-M. Jacquet. $\mu^2 Log$: Towards Remote Coordination. In P. Ciancarini and C. Hankin, editors, *Proceedings of the International Conference on Coordination Models and Languages*, Lecture Notes in Computer Science, pages 34–43. Springer-Verlag, 1996.
7. W. Fokkink. *Introduction to Process Algebra*. Springer-Verlag, 2000.
8. D. Gelernter and N. Carriero. Coordination Languages and Their Significance. *Communications of the ACM*, 35(2):97–107, 1992.
9. D. Harel. Statecharts: a Visual Formalism for Complex Systems. *Science of Computer Programming*, 8, 1987.
10. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
11. J.-M. Jacquet and K. De Bosschere. Blackboard Relations in the μlog Coordination Model. *New Generation Computing*, 19(1):23–56, 2001.
12. J.-M. Jacquet, K. De Bosschere, and A. Brogi. On Timed Coordination Languages. In A. Porto and G.-C. Roman, editors, *Proceedings of the 4th International Conference on Coordination Languages and Models*, volume 1906 of *Lecture Notes in Computer Science*, pages 81–98. Springer, 2000.
13. I. Linden and J.-M. Jacquet. On the Expressiveness of Absolute-Time Coordination Languages. In R. De Nicola, G.L. Ferrari, and G. Meredith, editors, *Proceedings of the 6th International Conference on Coordination Models and Languages*, volume 2949 of *Lecture Notes in Computer Science*, pages 232–247. Springer, 2004.
14. I. Linden, J.-M. Jacquet, K. De Bosschere, and A. Brogi. On the Expressiveness of Relative-Timed Coordination Models. *Electronical Notes in Theoretical Computer Science*, 97:125–153, 2004.
15. I. Linden, J.-M. Jacquet, K. De Bosschere, and A. Brogi. On the Expressiveness of Timed Coordination Models. *Science of Computer Programming*, 61(2):152–187, 2006.
16. T.M. Malone and K. Crowston. The Interdisciplinary Study of Coordination. *ACM Computing Survey*, 26(1):87–119, 1994.
17. R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, New York, 1980.
18. R. Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.
19. A.L. Murphy, G.P. Picco, and G.-C. Roman. Lime: A Coordination Model and Middleware Supporting Mobility of Hosts and Agents. *ACM Transactions on Software Engineering*, 15(3):279–328, 2006.
20. A. Omicini and F. Zambonelli. Tuple Centres for the Coordination of Internet Agents. In *Proceedings of the SAC Conference*, pages 183–190, 1999.
21. A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors. *Coordination of Internet Agents: Models, Technologies, and Applications*. Springer, 2001.
22. G.A. Papadopolous and F. Arbab. Coordination Models and Languages. *Advances in Computers*, 48, 1998.
23. V.A. Saraswat. *Concurrent Constraint Programming Languages*. The MIT Press, 1993.

Ubiquitous Sensing: A Prerequisite for Mobile Information Services

M.J. O'Grady[1], G.M.P. O'Hare[1], N. Hristova[2], S. Keegan[2] , C. Muldoon[2]
[1]Adaptive Information Cluster (AIC), University College Dublin (UCD), Belfield, Dublin 4, Ireland.
[2] PRISM Laboratory, University College Dublin (UCD), Belfield, Dublin 4, Ireland.
{M.J.OGrady, Gregory.OHare}@ucd.ie
{Natalya.Hristova, Stephen.Keegan, Conor.Muldoon}@ucd.ie

Abstract. Fundamental to the attainment of the ubiquitous computing vision is the capture, interpretation and intelligent use of diverse sensed data. This data may be dispersed throughout the environment, conform to different standards and, crucially, may be compromised in its availability and accuracy. Thus, obtaining a picture of the prevailing situation, or context, at a given time may prove problematic. To address this problem, the ubiquitous sensing concept is reconsidered. In particular, the potential of intelligent agents as an enabling technology for realizing distributed effective ubiquitous sensing applications is examined.

Keywords: Ubiquitous Sensing, Context-aware computing, Ubiquitous computing, Ambient Intelligence, intelligent agents.

1 Introduction

Mobility is an essential component of people's everyday lives, and in some case, it is an intrinsic feature of their work and leisure activities. This observation has motivated a number of computing paradigms aimed at meeting the needs of the mobile computing community – ubiquitous computing [1], pervasive computing [2] and wearable computing [3] are three well known examples. It is unlikely that any of these paradigms will become predominant; rather principles from each will be harnessed, adopted and deployed according the nature of the application domain. For this to happen, there are still some formidable barriers that must be overcome.

A particular danger that can arise with technology is that it can, in various ways, become insular, and immune to developments in both associated areas, and in areas that are not obviously connected yet that could contribute much to the development of the technology in question. Such is the case with ubiquitous computing. Significant strides have been made in some of the technologies essential to its realisation, for example, wireless communications, sensor technologies, mobile devices and so on, yet the vision remains quite some way from realisation. In this paper, the concept of ubiquitous sensing is explored in some detail. The need for ubiquitous sensing in ubiquitous computing and smart environments has been acknowledged [4]. However, its effective realisation remains an outstanding research issue. For the purposes of this

discussion, ubiquitous sensing is associated with the capturing of sensed data, its subsequent interpretation, and its judicious use in applications.

This paper is structured as follows: In Section 2, the concept of ubiquitous sensing is defined and explored in some detail. Section 3 is concerned with the intelligent agent paradigm and how this can be harnessed in ubiquitous sensing. Section 4 presents three case studies that illustrate the harnessing, interpretation and use of ubiquitously sensed data. Some future research is outline in Section 5 after which the paper is concluded.

2 Ubiquitous Sensing

Fundamental to ubiquitous sensing is the requirement for a networked infrastructure of embedded sensors. These sensors may be embedded within the surrounding infrastructure, as per Weiser's vision, or they may be mobile. Example of mobile sensors could include wearable devices or in-vehicle telemetric devices, to name just two. Their potential should not be under-estimated. A valid criticism of the original ubiquitous computing concept was that not all physical environments would ever be endowed with the necessary sensing infrastructure that ubiquitous computing demands. Such environments are not limited to exotic examples such as the polar regions; rather, they include rural areas that may be sparsely populated, yet the inhabitants still need access to various services. The IST objective "broadband-for-all" is an illustration of an international initiative that was conceived with the aim of bringing a technological service to less developed, peripheral and rural regions. Thus, it must be acknowledged that progress is being made to remedy deficiencies in existing infrastructure; and that as ubiquitous computing goes mainstream, it is likely that further international and national initiatives may be fostered to address outstanding infrastructural needs. However, while research on the basic sensing technologies, amongst others, continues, a particular research challenge concerns the harnessing of information from sensors and its meaningful interpretation. A classic example of this may be found in the context-aware computing paradigm.

2.1 Context-aware Computing

Context-aware computing [5] involves the harnessing of an individual's prevailing context and using it as a parameter for customizing or tailoring an application to their needs at that particular instance in time. Location is of course the classic example of context but there are others. However, which elements are of relevance will be influenced by the application domain in question. Capturing the pertinent elements of context may give rise to significant difficulties from a software engineering perspective. This difficulty is accentuated when it is considered that the required context may need to be harvested from sensors that are spatially distributed, and may use different communications protocols. This is a critical difficulty in the effort to realise ubiquitous sensing, and hints at the necessity of a software approach that is distributed and collaborative.

Assuming that the contextual elements have been captured, the next challenge that arises concerns the interpretation of these elements. Or more specifically, how meaning is ascribed to a particular combination of contextual cues. This is a non-trivial problem. Indeed, a degree of intelligence may well be required to make sense of contextual state prevailing at any given time. Even then, a confidence factor may need to be associated with the resultant contextual state.

A third challenge concerns the actual use of the contextual elements. Ideally, after expending significant effort in capturing the contextual elements and interpreting them, the application or service would harness these in such a way as to make a notable difference to the end user experience. This may, or may not be the case. Again, it may be necessary to harness intelligence techniques so as to identify how the known contextual state might be fruitfully harvested. One initiative that does this is Ambient Intelligence (AmI).

2.2 Ambient Intelligence

Ambient Intelligence (AmI) [6] was conceived by the Information Society Technologies Advisory Group (ISTAG) in the early years of this decade. The motivation behind its conception was the realisation that ubiquitous computing environments ran a serious risk of becoming unusable, as the number of artefacts and users increased. Specifically, it was envisaged that these artefacts would all be canvassing and competing for end-users' attention, resulting in an environment that would quickly be perceived as not being user-friendly. Thus, AmI envisages the adoption of intelligent techniques, and in particular, intelligent user interfaces, as a means of mediating between the environment and its inhabitants.

Not surprisingly, AmI does not ratify any particular intelligent technique or approach. Rather, it just acknowledges the necessity for such techniques to make ubiquitous computing environments habitable and productive. Thus while AmI encompasses a suite of technologies including ubiquitous computing, context-aware computing and so forth, it is primarily concerned with Human Computer Interaction (HCI) issues.

2.3 Realising Ubiquitous Sensing

Having considered some of the key elements in ubiquitous sensing, the necessary characteristics of an enabling software paradigm can now be considered.

- Distributed: Determining the prevailing context at any moment in time, except in the simplest cases of course, requires that context elements be harvested from a diverse range of sensors that are invariably spatially distributed.
- Collaborative: Though a sensor can capture some physical contextual element quite easily, it may need further information to specify a contextual state with a degree of certainty. Such information may well be available from another sensor in the network, or possibly, from a database on a fixed network node. Thus determining context is frequently not a singular or solitary activity; rather, it is an essentially

collaborative endeavour where sensor network nodes must share information if they wish to obtain a fuller picture of the prevailing contextual situation.

– Intelligent: Reconciling sensor data such that a contextual situation can be disambiguated may be exceptionally difficult in some cases. It may be necessary to adopt Artificial Intelligent (AI) techniques in an effort to identify the contextual state with some confidence. Finally, should it be necessary to render a user interface, either of the classic or multimodal variety, it may be also necessary to construct an intelligent user interface.

Having identified the essential characteristics of ubiquitous sensing, we can now look at a paradigm that encapsulates these characteristics: Intelligent Agents.

3 Intelligent Agents

Intelligent Agents are a mature and practical realization of Distributed Artificial Intelligence (DAI) [7]. What constitutes an agent is subject to interpretation but agents incorporate some combination of autonomy, mobility, reactivity, proactivity and sociability, amongst others. During design, these attributes will be harnessed by the software engineer as the application domain demands.

Fundamental to the notion of agency is that of a multi-agent system (MAS). Agents rarely exist in isolation; rather, they contribute to an agent community that has been formed, either a priori or in an ad-hoc manner, to solve a particular problem. Essential to this is their social or communicative ability, and implicitly, the availability of an agreed ontology. When ubiquitous sensing is considered, one can easily envisage a network of sensors, each assigned an individual agent. These agents can communicate and determine the status of sensors in their vicinity, and thus construct a more complete model of the prevailing context than could be obtained by the agents acting unilaterally.

Given the emphasis on intelligence in the previous discussion, it is appropriate to comment further on this. The characteristics of agents documented above satisfy the basic criteria of agenthood. From an AI perspective, something further is expected, and this concerns the implicit assumption that the agent adopts a sophisticated reasoning ability that enables the agent act in a rational manner. A number of models exist, but for the purposes of this discussion, the BDI [8] model will be discussed briefly.

3.1 The BDI Model

Belief-Desire-Intention (BDI) represents one interesting interpretation of how agents might reason. Clearly, the model is based on what may be termed *mentalistic* concepts, and in this way, it was hoped that this would represent a more intuitive approach to reasoning. Unlike, other proposed models, it is computationally tractable, even on devices of limited capacity as will be seen in the next section. BDI agents follow a *sense-deliberate-act* cycle. A model of their world is constructed as a belief

set. This model is deliberated on in light of the agent's desires. Desires represent those courses of action that the agent would like pursue, given a certain state of their world, or in other words, *context*. At any given time, it may be that the prevailing contextual state is such that the agent can pursue one or more courses of actions. These courses of action are referred to as intentions in BDI parlance. Once a cycle has been completed, the agent updates their belief set and proceed as before. Though the BDI approach is computationally expensive, ongoing developments have enabled its deployment on lightweight devices such as mobile phones. In the next section, one pioneering platform that achieves this objective is described.

3.2 Agent Factory Micro-Edition (AFME)

Agent Factory Micro Edition (AFME) [9] is a minimised footprint agent platform specifically designed for resource constrained mobile devices. It is loosely based on Agent Factory [10], a pre-existing agent platform for personal computers. It uses a subset of the Agent Factory Agent Programming Language (AFAPL) and augments it with a number of features specific to AFME. AFAPL is founded on a logical formalism of belief and commitment. Rules that define the conditions under which agents should adopt commitments are used to govern and encode agent behaviour.

An AFME platform comprises a scheduler, several platform services, and a group of agents. The scheduler is responsible for the scheduling of agents to execute at periodic intervals. Rather than each agent creating a new thread when they begin operating, agents share a thread pool.

In AFME agents are rational; sometimes they collaborate other times they compete depending on the context. In certain circumstances the developer will want to ensure that agents always behave in a particular manner. This type of behaviour is modelled, within AFME, through the use of fixed utility values that ensure that under such circumstances it is always in an agent's interest to behave in a certain fashion. AFME has been successfully harnessed to realise a number of applications in the ubiquitous sensing domain and these are now described.

4 Case Studies in Ubiquitous Sensing

In this section, three case studies are presented. Each one adopts the agent paradigm and uses agents to intelligently interpret the user's context such that the corresponding service is delivered in those particular circumstances when the user is in most need of, or is most receptive to it. In each case, the AFME platform has been harnessed.

4.1 Ubiquitous Sensing in the Tourist Domain

Supporting mobile tourists in their leisurely pursuits is the main objective of Gulliver's Genie [11]. The Genie is typical of a number of systems in the objectives it seeks to achieve; however, the one of most similarity is CRUMPET [12] as this also adopts the agent paradigm, though its use of agents is quite different.

Gulliver's Genie supports the information needs of tourists. It provides a map that shows their spatial context, and on encountering tourists attractions, proceeds to present multimedia presentations about that attraction to the tourist. It can be hosted on a PDA or such class of device. Architecturally, the Genie comprises a suite of agents that have been deployed on tourists' devices and on a fixed network server, and all collaborate to deliver the required service. Two agents are deployed on the PDA:

- A Spatial Agent monitors the tourist's spatial context via a GPS sensor. It interprets the measurements, derives behaviour and movement models, and ensures that supporting suite of agents on the fixed network is always kept updated.
- A Cache Agent which ensures that the cache on the PDA always contains information that is most pertinent to the tourist's current location. To achieve this objective, it is dependent on information from the other agents in the MAS.

Supporting the tourist community is a suite of agents that reside on a fixed networked host. These agents communicate with those on the PDA via a wireless connection, in this case either 3G or GPRS. The most important these agents are as follows:

- Tourist Agent: This agent acts as the tourist's surrogate. All communications from the PDA must pass to this agent. It coordinates with the other agents to deliver any information required by the agents on the PDA.
- GIS Agent: This agent uses a pre-constructed model of the tourist's environment to identify what attractions the tourist is expected to encounter, if they continue their current trajectory.
- Profile Agent: All information is personalised to the tourist personal profile and cultural interest model. The Profile Agent maintains a model for each tourist registered with the Genie, and provides individual models to the other agents on request. In particular, the agent is responsible for dynamically updating the tourist's model in response to their activities.
- Presentation Agent: Dynamically constructing multimedia presentations is the task of this agent. In doing this, it is completely dependent on cues from the Cache Agent on the PDA, as well as both the Profile and GIS agents on the fixed networked node.

Architecturally, the agents form a kind of middleware that interprets the tourist context such that information can be intelligently identified, filtered and constructed into a meaningful presentation that is in harmony with the tourist's position, activity and information requirements. In doing way, they fulfil the requirements identified in Section 2.3 for achieving a ubiquitous sensing solution. The Genie is of necessity an inherently distributed system. No single in agent can deliver the services offered by the Genie on it own; rather, an agent must collaborate with a series of agents to achieve the overall objectives of the Genie. Finally, the disjoint information must be intelligently analysed and harmonised such that it meets the tourists' information needs and expectations.

4.2 Ubiquitous Sensing in the U-commerce Domain

Easishop [13] is a personalised mobile shopping system. By permitting shoppers to build and maintain shopping profiles, suitable product offerings can be delivered, negotiated and resolved in an appropriate context. An m-commerce implementation, Easishop utilises a specially designed user-interface, optimised for handheld mobile devices to allow the input of preferences and profile-specific data like age, gender and interests. The shopper communicates silently and without explicit input with stores as they are encountered. The architecture is completed by a centralized server where a competitive process of auctioneering may occur. The system is founded on a collection of complementing principles.

Competitiveness through Reverse Auctioneering

A core objective of Easishop is to deliver competitive product offerings to users in an appropriate context (figure 1). One effective mechanism to deliver this is to provide a domain in which open, reverse-auctioneering occurs. In this process, shoppers publicise which products are sought. At this point, any number of interested merchants may propose offerings, any of which the shopper may accept or reject at will. To deliver this structure in a secure, efficient and effective fashion, a three-tiered architecture has been devised. These are i) device; ii) hotspot and iii) marketplace. A brief description of each is described below:

- Device: This tier embodies a suite of software installed on supported hardware like PDAs and smartphones. The role of the software is two-fold – i) to enable the user to create, maintain and remove shopping preferences and user profiles; ii) to govern active intermittent communication with detected hotspots as the user passes through an active shopping zone. A degree of separation is maintained between these two modules. It has been found that this approach encourages easier scalability, interoperability and modularity.
- Hotspot: Each participating store in Easishop has special hardware installed in the shop front area. This hardware is installed with supportive Easishop software and, together, these are termed the Easishop hotspot. The primary function of the hotspot is to act as a conduit between the device and the marketplace, in permitting representative agents from the device to move from the device through the hotspot and onto the marketplace, before initiating a reverse auction. A secondary function of the hotspot is to allow the passage of a set of rules, embodied in a special type of software agent, to be passed from the store to the marketplace. It is upon this set of rules that a bid can be prepared for consideration by the user in the reverse auction process.
- Marketplace: The marketplace forms the backbone of the Easishop architecture. Multi-threaded processing permits multiple simultaneous real-time reverse auctioneering as well as product catalogue dissemination. To deliver the required functionality, the marketplace:
  i) maintains and distributes a master catalogue of products. This is an ontology of all products active within the system. Only relevant information is disseminated – i.e. only updates pertaining to products stocked by a particular store are passed to that store. Likewise, only product data of interest to a particular user is forwarded to that user;

ii) governs the auctioneering process. This involves ensuring that all participants of an auction observe the rules of the auction process;
iii) regulates the movement of agents throughout the system. This involves coordination of movement to ensure that agents, for example, upon completion of an auction, are returned to the device of origin.

Fig. 1. Constructing a shopping list with Easishop.

Distributed Persona Realization through Mobile Software Agency

At every point of the Easishop architecture, software agents, some mobile, others stationery, are employed. The agent, in terms of Easishop, is a collection of software-defined rules and behaviors which together determine how the agent will respond in any given situation. The most pertinent example of this are the agents that are used to represent buyers (shoppers) and sellers (stores). A particular shopper agent may express a preference for long-sleeved shirts over the short-sleeved variety, for example, while a particular store may wish to reduce all prices by 10 percent on Saturdays. In this way, the agents are said to be persona-realizing. It is necessary that the bulk of these agents be mobile, in that they are free to move between device, hotspot and marketplace as the need arises. It should be noted that some aspects of these three tiers infer notably different environmental characteristics and those agents that are mobile are designed to operate effectively regardless of which tier they happen to be situated.

Location awareness through Proximity Detection and Inter-tier Communication

A central tenet of the Easishop approach is to provide product offerings to the user in context. The most obvious aspect of context in this regard is the location of the shopper and the location of the stores (the location of the marketplace is irrelevant).

Furthermore, the relationship between these two locations (e.g. distance) is of particular relevance. The primary mechanism by which the location of the shopper is deduced is through Bluetooth proximity detection. This inherent characteristic of the Bluetooth standard (by which effective radio range extends only to approximately 50 metres) is exploited to allow an accurate and dynamic location fix for any given shopper. This location information is passed as detected by the hotspot to the marketplace and in this way relevant communication between the tiers can be coordinated. Context determination for the hotspots is obviously a simpler prospect as the location of each is static.

4.3 Ubiquitous Sensing for Opportunistic Advertising

Ad-me (ADvertising for the Mobile E-commerce user) [14] provides location-aware and dynamic information, interactive services and personalization. It operates in an outdoor environment and obtains user location using a GPS receiver, connected to a PDA. Ad-me deploys Multi-Agent techniques, adapting its behavior to users' emotions and delivering targeted advertisements to users when they need them (temporal context), where they need them (spatial context) and how they need them (device context). The Ad-me client software module is implemented as a stand-alone java application. The motivation for this was enhanced portability; greater efficiency and enhanced control over the Graphical User Interface.

Ad-me addresses the primary objective of investigating whether the knowledge of the user's emotional state can increase the effectiveness of wireless advertising. In particular, this includes:

- determining which emotions a Wireless Advertising (WA) system should be concerned about;
- determining what should be the response of a WA system that knows the user's emotional state;
- and investigating the feasibility of using biometric methods for measuring the user's emotions in the mobile advertising field.

Scientific findings suggest that emotion plays a significant role in producing rational behavior and rational decision-making [15]. In order to enable computers (and mobile devices) to recognize the emotional cues from the user, it is necessary to record specific autonomic response signals from non-invasive bio-sensors that can be used in conjunction with a wearable computer for real-time portable signal acquisition [16]. Measurements like Blood Volume Pulse (BVP), heart rate (EKG), and galvanic Skin Conductance (SC) are commonly used in emotion research experiments. Characteristic patterns of these signals have been found which correlate with different self-reported emotional states. The most widely accepted axes for the categorization of emotions are *valence*: the discrimination between positive and negative experiences; and *arousal*: the intensity with which the emotion is experienced. These two axes have been widely accepted in many diverse theories and research [17]. Various experiments have demonstrated that it is possible to sense someone's emotions via such bio-sensors. To detect a user's emotion, we use the ProComp2 encoder device, augmented with BVP and SC sensors. The SC indicates the level of

arousal and the BVP indicates the valence. An IPAQ PDA interfaces the ProComp2 encoder through its COM port via a custom interface for the sensor-to-PDA connection.

Ad-me comprises a federation of agents. The majority of the agents and other system components are located on server side. On the user's PDA, a *Client Agent* receives data from both the SC and the BVP sensors, via the Procomp2 device, and establishes the baseline of the SC and the BVP signal for the particular user. The agent monitors the SC and BVP values for changes, and on exceeding a predefined threshold, the measurements are reported to the server.

Decision networks, an extension of the Bayesian networks, are employed. Such networks have already been applied in a similar scenario [18]. From a node perspective, decision networks contain nodes representing probabilistic events in the world, nodes representing agent choices (decision nodes), and nodes representing the agent's utility function. Our Decision diagram is depicted in Figure 2 where the agent shows an advertisement when the user is relaxed, or joyful, updates user profile if the user experiences excitement, and discards the advertisement if it detects anger or frustration.

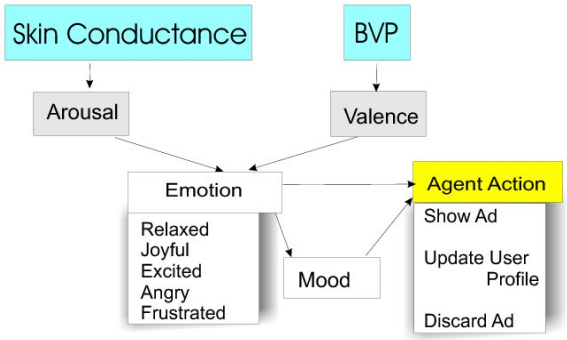


Fig. 2. Decision Diagram for Ad-me.

The agents on the server side include:

- the User Profiling Agent that makes inferences and creates and stores a profile for the given user into the database, using a completed questionnaire about their interests and the type of device they are using.
- the Context Agent that generates a set of beliefs about the user and their device context concerning the demand and need to display content for that particular user.
- the Find Objects Agent that finds all the requested (push or pull) objects situated within the user's vicinity and calculates the value of the relation, called *interest*, between the user and each of these objects.

An Advertising Agent residing on the server correlates the emotional data together with the other aspects of the user's context to determine when it is appropriate to place an advertisement. On presenting an advertisement, the Emotion Agent (on the PDA) stores a one-minute record of the user emotive reaction and sends it to the User

Profiling Agent (on the server) which proceeds to update the user's profile. In the case where the Emotion Agent inferred that the user has got irritated while viewing an advertisement, Ad-me immediately discards the advertisement. An Advert-sales Agent monitoring agent records and analyzes advertising and sales data.

5 Ongoing Research

Ongoing projects at present include the development of a mobile blogging application that enables mobile subscribers to record and retrieve blogs in mobile contexts, using standard handsets. It is hoped that this project will provide some useful insights into the potential of agents for realizing intelligent user interfaces on mobile devices. A second project concerns the deployment of a suite of agents for monitoring a network of PTZ cameras and their effectiveness at critical scene recognition and analysis.

While the deployment of the intelligent agent paradigm on archetypical mobile devices represents a significant milestone in the process of enabling ubiquitous sensing at the periphery of telecommunications networks, and to a lesser extent, the internet, sensors represent a further and most significant challenge. Indeed, this challenge must be conquered if ubiquitous computing and AmI are to become essential facets of everyday life. As an initial step towards this, the AFME platform is currently being deployed on the Crossbow Stargate platform. Strictly speaking, the Stargate may be regarded as a base station or a data sink in a traditional WSN topology. However, it represents a realistic intermediate platform between the devices of the mobile telephone genre and sensor nodes. It is envisaged that the subsequent process of deploying AFME on sensor nodes such as the IMote2 and the SunSPOT can then commence.

6 Conclusion

Ubiquitous sensing is an implicit yet essential lynch-pin of ubiquitous computing and Ambient Intelligence. Indeed, it is an essential prerequisite if the myriad of applications and services promised by the proponents of ubiquitous computing are to materialise. In this paper, it has been argued that the effective realisation of ubiquitous sensing requires a software solution that is inherently distributed, collaborative and intelligent. The intelligent agent paradigm is one illustration of an approach that encapsulates all these features. To illustrate the viability of the approach, three mobile computing scenarios that harness intelligent agents have been described.

Acknowledgments. This material is based upon works supported by the Science Foundation Ireland (SFI) under Grant No. 03/IN.3/1361.

References

1. Weiser, M.: The Computer for the Twenty-First Century. Scientific American, 265 (3) (1991), 94-100
2. Satyanarayanan, M.: Pervasive computing: Vision and challenges. IEEE Personal Communications, 8, (2001) 10-17.
3. Rhodes, B.J., Minar, N. Weaver, J.: Wearable Computing Meets Ubiquitous Computing: Reaping the Best of Both World. Proceedings of the Third International Symposium on Wearable Computers, San Francisco, CA, USA. (1999) 141-149.
4. Essa, I.A., Ubiquitous sensing for smart and aware environments, IEEE Personal Communications, 7 (5) (2000) 47-49
5. Dourish, P.: What we talk about when we talk about context. Personal & Ubiquitous Computing, 8 (2004) 19-30
6. Vasilakos, A., Pedrycz, W. (Eds): Ambient Intelligence, Wireless Networking, Ubiquitous Computing, Artec House. (2006)
7. Foundations of Distributed Artificial Intelligence (G.M.P. O'Hare and N. Jennings eds), John Wiley and Sons, Inc. (1996)
8. Rao, A.S, Georgeff, M.P.: Modelling Rational Agents within a BDI Architecture. Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann Publishers, San Mateo, CA. (1991) 473-484
9. Muldoon, C., O'Hare, G.M.P., Collier, R. O'Grady, M.J.: Agent Factory Micro Edition: A Framework for Ambient Applications. Proceedings of Intelligent Agents in Computing Systems Workshop (held in Conjunction with International Conference on Computational Science (ICCS)) Reading, UK. Lecture Notes in Computer Science (LNCS), Vol. 3. Springer-Verlag Publishers (2006) 727-734
10. Collier, R.W. O'Hare, G.M.P. Lowen, T. Rooney, C.B.F.: Beyond Prototyping in the Factory of Agents, Proceedings of the 3rd International Central & Eastern European Conference on Multi-Agent Systems (CEEMAS'03), Prague, Czech Republic. LNCS Vol. 2691. Springer-Verlag (2003) 383-393.
11. O'Grady, M. J. O'Hare, G. M. P.: Just-In-Time Multimedia Distribution in a Mobile Computing Environment. IEEE Multimedia, 11(4), (2004) 62-74
12. Poslad, S., Laamanen H., Malaka R., Nick A., Zipf, A.: Crumpet: Creation of user-friendly mobile services personalised for tourism. Proceeding of the Second IEE International Conference on 3G Mobile Communication Technologies, London, UK. (2001)
13. Keegan, S., O'Hare, G.M.P.: Easishop: Enabling uCommerce through Intelligent Mobile Agent Technologies. In: Proceedings of 5th International Workshop on Mobile Agents for Telecommunication Applications (MATA'03), Marrakesh, Morocco. LNCS Vol. 2881. Springer-Verlag (2003) 200-209.
14. Hristova, N., O'Hare, G.M.P.: Ad-me: Wireless Advertising Adapted to the User Location, Device and Emotions. In: Proceedings Software Track of the Thirty-Seventh Hawaii International Conference on System Sciences (HICSS-37). IEEE Computer Society Press. (2004)
15. Picard, R.W., Affective Computing, MIT Press, Cambridge, (1997)
16. Starner, T., Mann, S., Rhodes, B., Levine, J., Healey, J., Kirsch, D., Picard, R., Pentland, A.: Augmented Reality through Wearable Computing. Presence, 6(4) (1997) 386-398
17. Lang, P.J.: The emotion probe: Studies of motivation and attention. American Psychologist. 5 (50), (1995) 372–385
18. Prendinger, H., Mori, J., Mayer, S., Ishizuka M.: Character-based interfaces adapting to users' autonomic nervous system activity. In: Proceedings of the Joint Agent Workshop (JAWS-03). Awaji, Japan. (2003) 375–380

A Unified Authentication Solution for Mobile Services

[1]Håvard Holje, [2]Ivar Jørstad & [3]Do van Thanh

[1] Norwegian University of Science and Technology, O.S. Bragstads plass 2, NO-7491 Trondheim, Norway, holje@stud.ntnu.no
[2]Ubisafe AS, Bjølsengata 15, NO-0468 Oslo, Norway, ivar@ubisafe.no
[3]Telenor R&I - Norwegian University of Science and Technology, Snarøyveien, NO-1331 Fornebu, Norway, thanh-van.do@telenor.com

Abstract. Current mobile phone architecture does not provide adequate security support for applications. This paper contributes to the opening of the mobile terminal architecture by presenting a solution that enables non-telephony applications to make use of the strong authentication functions located on the SIM card. The solution is based around a supplicant which provides services with access to the native authentication mechanisms of a GSM/UMTS network. All communication between the supplicant and the network side is performed using standards protocols.

Keywords: mobile service, authentication, SIM, mobile handsets, Java, web services

1 Introduction

From a simple device terminating the mobile network, the mobile phone has evolved to become a quite advanced device capable of hosting applications that are until now run only on stationary computers. The limitations in terms of processing, storage and battery life are considerably reduced, and the mobile phone will soon become a mobile computer. However, there is one major obstacle, which is the current closed architecture of the mobile terminal. Indeed, the architecture is very much telephony centric, i.e. it is built to support the traditional telecommunication services like GSM voice, SMS, WAP, etc. Other applications like browsing, Web services, P2P applications get very little support and in most cases have to manage by themselves.

This paper contributes to the opening of the mobile terminal architecture by presenting a solution that enables non-telephony applications to make use of the strong authentication functions located on the SIM card.

The SIM card, which is a tamper resistant Smart card, utilizes ISO-standardized Application Protocol Data Units (APDU) to communicate with host devices via PIN codes and cryptographic keys.

Existing (strong) authentication schemes on mobile handsets suffer of serious drawbacks. Some are completely separated from the SIM and require additional

elements such as a Smart Card, a one-time password generator, etc. The others access the SIM authentication functions indirectly via SMS. The paper starts with a summary of related works. The Unified Authentication is then presented thoroughly. All the components are described in detail.

2 State-of-the-art in authentication on mobile phones

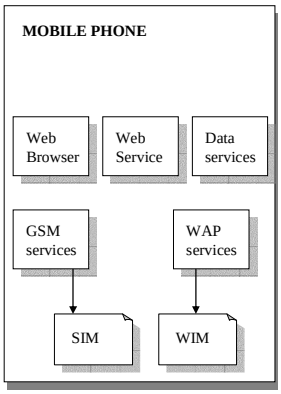


Figure 1. Mobile phone service architecture

The mobile phone is originally intended only for voice communication service, i.e. telephony. The mission of the SIM card is to carry the subscriber's identity and to provide security functions like encryption and authentication. In fact, the SIM card is acting as a slave executing order only from the GSM services which include telephony and SMS (Short Message Service) as shown in Figure 1.

The GSM network authenticates the identity of the subscriber through the use of a challenge-response mechanism. The GSM security model is based on a shared secret between the SIM and the AuC (Authentication Center) of the subscriber's home network. The shared secret (Ki) is a unique 128 bit key.

The authentication is initiated by the fixed network and it is based upon a simple challenge-response protocol. First the network identifies the MS (Mobile Station) by retrieving the IMSI (International Mobile Subscriber Identity). Next the MSC (Mobile Switching Center) contacts the MS's HLR (Home Location Register), and asks it to send a triplet containing RAND, SRES' and Kc. The triplet is computed by the AuC, which is the only entity in the GSM network knowing Ki besides the MS itself, and it is sent back to the MSC by the HLR.

When the MS retrieves the triplet, it will use the built-in authentication mechanisms in the SIM to generate a signed response (SRES). This is the unique part of the GSM authentication scheme, which makes it so strong. The private key is never

sent over the network. Instead the MS computes its own SRES by feeding the COMP128 algorithm with the local version of Ki and the newly retrieved RAND.

When the mobile phone evolves other services start to appear. Unfortunately, they usually do not have access to the security functions on the SIM card, but have to implement their own solutions. As shown in figure 1 there is no connection between the SIM card and emerging services like Web browser and Web services.

2.1 Authentication for WAP-based Services

For WAP (Wireless Application Protocol) application a WAP Identity Module (WIM) [1] is defined and used in performing WTLS (Wireless Transport Layer Security) [2] and application level security functions, and especially, to store and process information needed for user identification and authentication. The WIM functionality can be implemented on a smart card. A smart card implementation is based on ISO 7816 [3] series of standards. The WIM is defined as an independent smart card application, which makes it possible to implement it as a WIM-only card or as a part of multi-application card containing other card applications, like the GSM SIM.

2.2 Authentication for Web-based Services

For Web-based services accessed through a Web browser, stronger authentication is offered by using the One-Time-Password scheme. However, this solution is not the ideal one for mobile phone. The OTP must be generated by a device, which the user must bring along, or it can be sent to the user via SMS. Anyway, the user has to enter in the OTP manually, in addition to username and password, which might be a complicated procedure on small mobile handsets with poor keyboards.

Another option is to use a PKI-solution, which requires a PKI client installed in the SIM card as separate application. To carry out authentication, the Web site has to send challenges to the PKI Client using SMS as carrier. Only when the authentication is successful, the Web server will return to the mobile browser. Quite often, the browsing session has been terminated following of the termination of the data packet session, e.g. GPRS, UMTS.

2.3 Authentication for Java-based Services

The Java 2 Platform, Micro Edition (J2ME) is a Java platform optimized for small devices with limited memory and processing power, such as mobile phones and PDA's. J2ME is divided into configurations, profiles and optional packages. Devices need a configuration adapted to their processing capabilities and the profile implements higher-level APIs that further define the application life-cycle model, the user interface, persistent storage and access to device-specific properties.

For J2ME applications there is recently defined the Security and Trust Services API (SATSA/JSR177) which extends the security features for the J2ME platform, through the addition of cryptographic APIs, digital signature service, and user credential management. SATSA also defines methods to communicate to a Smart Card, by leveraging the APDU protocol. The SATSA spec says there is no Smart Card access for untrusted (unsigned) MIDlets. Hence one has to sign the MIDlet with a certificate issued by the operator or the manufacturer, to be able to connect to the SIM.

JSR-248 (Mobile Service Architecture), which defines the next generation Java platform for mobile handsets, mandates the support of SATSA-APDU when a security element exists on the device, i.e. a Smart Card or a SIM card.

With SATSA, the necessary security functions are offered to the J2ME applications but the architectural problem is still not solved. It is not simple for applications to make use of these security functions and there is no point to require that each application must integrate the security functions.

2.4 Related work

As far as the authors know there exists no similar solution to what is proposed in this paper. However, related work regarding SIM authentication do exist, but the existing solutions are either product specific or they are based on a regular PC and not a mobile phone. An example of a product specific solution is the built-in EAP-SIM supplicant provided on the Nokia 9500 communicator. It is a native application provided by the manufacturer and can be combined with the 802.1x framework to achieve strong SIM based authentication in WLAN's.

Another similar approach is the SIM Strong project, which aims to extend the use of GSM SIM authentication to internet Web Services. Telenor, Axalto, Linus and Oslo University College have implemented a proof-of-concept prototype together in Oslo [4]. The prototype demonstrates the possibility of implementing innovative services in a heterogeneous environment using Liberty Alliance Federation Standard [5].

3 The Unified Authentication solution

The goal of the Unified Authentication solution is to provide a unified authentication mechanism for any type of application and service on the mobile handset. In addition, there are the following requirements:

- The authentication mechanism must be strong
- The authentication mechanism must be mutual
- The authentication mechanism must be user-friendly
- The authentication mechanism should be simple to add to existing and future mobile services
- The authentication mechanism should be cost-effective to establish for service providers

The main idea is to utilize the fact that the GSM SIM is a tamper resistant Smart Card accomplishing the ISO8716 Smart Card specifications. By utilizing the existing GSM SIM authentication mechanisms for IP based services, we want to achieve strong two-factor authentication, without other user interaction than typing the PIN.

To ensure that the user is who he/she claims to be, the PIN function on the SIM must be enabled and the user have to provide a valid PIN to get access to the service. This mechanism prevents misuse if the mobile handset is stolen, since the SIM is blocked after 3 invalid PIN attempts.

As depicted in figure 2, the SIM Supplicant is located in the Mobile handset. The SIM Supplicant is an important component of the proposed system. It is responsible for all communication between the SIM, the browser and the Identity provider. To be able to get access to the SIM functions, the supplicant utilizes the Security and Trust Services API (SATSA) for J2ME. The SATSA-APDU package allows the supplicant to communicate with the SIM by leveraging the APDU protocol, which is an ISO7816 compliant low-level protocol for Smart Card communication.

When a user wants to access a service on a Mobile handset, he/she uses the browser to contact the Service Provider. If the service requires authentication, an authentication request is returned. The Supplicant is contacted and initializes the authentication procedure.

The SIM Supplicant provides the subscriber identity and valid user credentials to the Identity Provider. The Identity Provider performs a lookup of the user in an associated Authentication Server and uses the existing GSM authentication mechanisms to authenticate the user. The Authentication Server uses a GSM gateway to communicate with the GSM network during the authentication.

When the subscriber is authenticated against the Identity Provider, a security association (SA) is created and returned to the browser. When the Service Provider has verified the claimed authenticity, the user is authorized to use the requested service

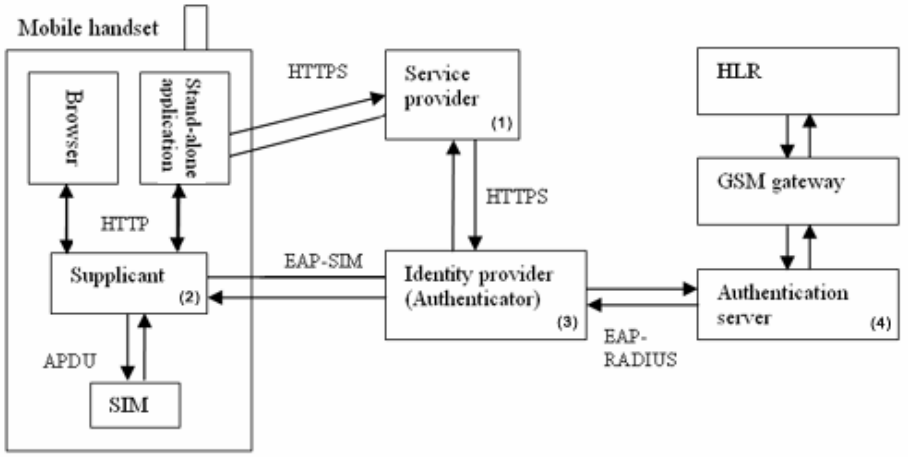


Figure 2. Overall architecture of generic SIM authentication system

3.1 Components

In this section the most important components in figure 2 will be explained in more details. The interface between the components is explained further in section 3.2

3.1.1 Service Provider (SP)

The Service Provider, component (1) in figure 2, offers services to the users and initializes the authentication procedure. When it receives a service request from a client, it responds with an authentication request to the local Supplicant on the mobile handset, if the client is not already authenticated. If the client provides an authentication token, the SP may contact the Identity Provider to verify the claimed authenticity. If the security association is valid, the SP authorizes the client to the requested service.

3.1.2 Supplicant

The Supplicant (2) is the major contribution to the proposed authentication system. It is a generic J2ME application acting as a local proxy on the mobile handset. The Supplicant provides an interface to the GSM SIM by utilizing the SATSA-APDU protocol. SATSA-APDU extends the security features for the J2ME platform, and makes it possible for the Supplicant to extract user credentials from the SIM.

The Supplicant is also implementing the EAP framework which makes it capable of exchanging EAP-SIM messages with the Identity Provider (Authenticator).

In an authentication sequence the Supplicant retrieves GSM authentication challenges from the Identity Provider by means of EAP-SIM messages. Next the Supplicant provides the GSM authentication challenges to the SIM and retrieves its user credentials by exchanging command and response APDUs, as defined by the ISO7816 Smart Card standard.

3.1.3 Identity Provider

The Identity Provider (3) is responsible for locating a suitable Authentication Server and it acts as an intermediary between the Supplicant, the Authentication Server and the Service Provider.

The Identity Provider translates EAP-RADIUS messages from the Authentication Server into EAP-SIM messages and passes it to the Supplicant. It is also storing information about authorized Supplicants, so the Service Provider can verify a Supplicant's claimed authenticity.

To get access to the Authentication Server, a RADIUS client must be implemented. A mutual trust between the Identity Provider and the Authentication Server is required.

3.1.4 Authentication Server

The Authentication Server (4) is performing the user authentication against the GSM network. It could be realized as a RADIUS server, which is the de-facto standard for remote authentication, but other Authentication Servers like DIAMETER may also be used. To perform the GSM SIM authentication, the RADIUS server will use the GSM gateway interface to contact the HLR of the subscriber.

3.3 Interfaces

3.3.1 SIM interface

To be able to communicate with the GSM SIM through the mobile handset it is required that the handset provides a SIM access interface. The SATSA-APDU package, described in section 2.3, provides such an interface. For more details regarding the SATSA-APDU API, we refer to SATSA Developer's Guide [6].

3.3.2 Authenticator interface

Extensible Authentication Protocol (EAP) [7] is a framework supporting multiple authentication methods. The Authenticator implements EAP-SIM [8] to communicate with the Supplicant, and EAP-RADIUS [9] to communicate with the Authentication Server. The EAP specification only discusses usage within a point-to-point protocol (PPP), which is a low layer protocol. The encapsulation of EAP messages in the higher layer protocols is not specified in the specifications. EAP is only dealing with authentication at the Application level.

Hence, there is a need to secure the communication channels between the components to ensure integrity and confidentiality. To solve this problem, EAP over TCP/IP may be chosen with SSL/TLS to maintain the integrity and confidentiality between the different components.

3.3.3 Supplicant interface

The Supplicant is a J2ME MIDlet running independent of the other applications on the mobile handset. It acts as an HTTP proxy and is listening to incoming request on a

local port. The interface between the Supplicant and the client application is defined by the XML schema in figure 5.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Supplicant">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element name="AuthenticationRequest">
          <xsd:complexType></xsd:complexType>
        </xsd:element>
        <xsd:element name="AuthenticationResponse">
          <xsd:complexType>
            <xsd:choice>
              <xsd:element name="Authenticated" type="xsd:boolean"/>
              <xsd:element name="SecurityAssociation" type="xsd:integer"/>
              <xsd:element name="ErrorDescription" type="xsd:string"/>
            </xsd:choice>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Figure 5. XML Schema defining the Supplicant interface

The defined Supplicant interface is very simple and consists of either an AuthenticationRequest or an AuthenticationResponse. If the authentication succeeds, a security association is returned inside the AuthenticationResponse. Otherwise, an error description will be returned.

The defined protocol provides a generic interface to the Supplicant, which means that any kind of client applications supporting HTTP can communicate with the Supplicant, with none or minor modifications.

3.3.4 Client application – Service Provider (SP) interface

The client application communicates with the SP by opening a standard secured HTTP connection. When the SP receives a service request, it checks whether the client is authenticated or not. If the client is not authenticated, the SP will respond with an authentication request. If the requesting client is a browser, the SP will redirect the authentication request to the local Supplicant on the mobile handset.

When the client is authenticated via the Supplicant, it requests the SP again, but this time it provides an authentication token (security association) together with the service request. The SP verifies the claimed authenticity and if everything is ok, the client is authorized to use the requested service.

3.3.5 Service Provider (SP) – Identity Provider (IDP) interface

The SP must be able to communicate with the IDP as well. When the client application is authenticated, the SP must be able to verify the claimed authenticity. The SP sends a request to the IDP via a secured HTTP connection, and the IDP responds with the corresponding security association. The communication channel between the SP and the IDP must be secured with SSL/TLS and both parties must be

authenticated to each other. I.e. they have to exchange certificates to each other before they can communicate.

3.3 Design choices

We chose J2ME as platform for the SIM Supplicant to reach as many users as possible, since most of the handheld devices implement a J2ME runtime environment. One major challenge when developing for the J2ME platform is the closed architecture of the mobile terminal. It is difficult to get access to security functions like SIM access and other native features. But the SATSA-APDU package makes this possible

Another obstacle is the communication between the local Supplicant and the mobile browser. In principle not-native applications does not have access to other native applications because of the closed architecture. But since the mobile terminal is opening up, e.g. with the MIDP 2.0 Push registry [10], which enables MIDlets to set themselves up to be launched automatically without user initiation, it is just a matter of time before most of the mobile handsets will act as a small PC. The new Mobile Services Architecture (MSA) [11] defines the next generation of the Java platform for mobile devices, which will make it easier to develop applications for a broad range of devices.

3.4 Sequence diagram of successful authentication

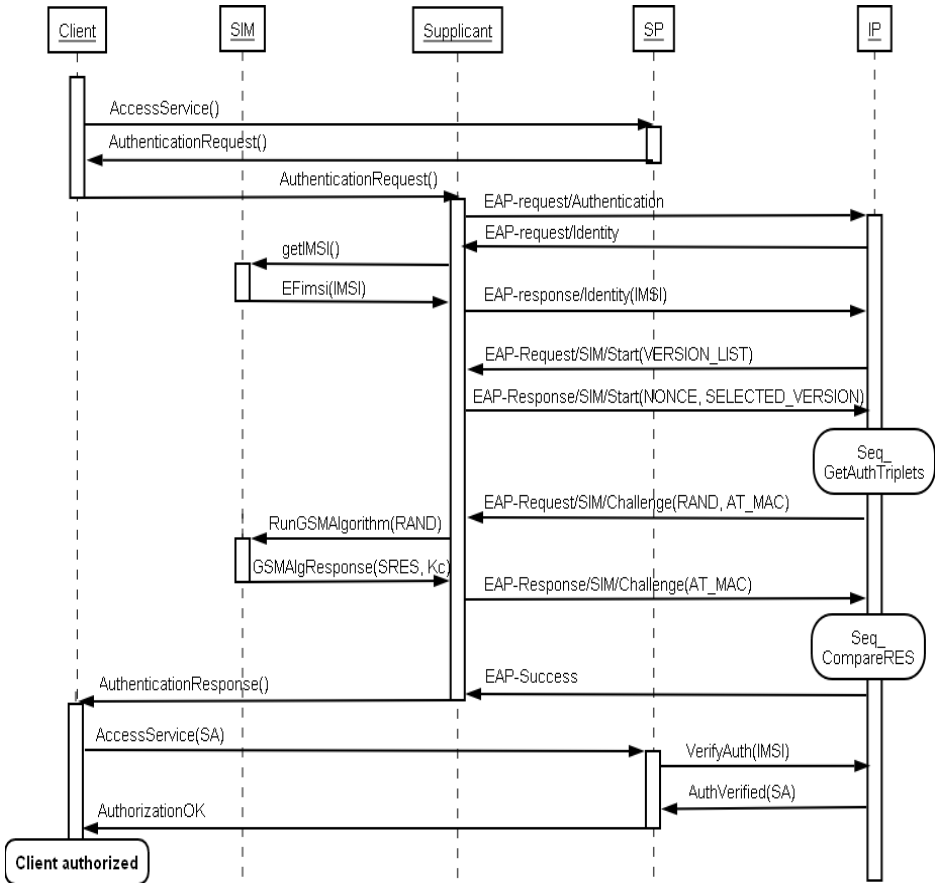


Figure 3. Sequence diagram for successful authentication of a subscriber

Figure 3 shows a sequence diagram for a successful authentication procedure. The client requests a service from the Service Provider (SP) and the SP responds with an AuthenticationRequest if the client doesn't provide a valid authentication token. The AuthenticationRequest is redirected to the Supplicant, which is responsible for authenticating the client against the Identity Provider (IDP). The IDP utilizes the existing GSM mechanisms for authenticating the client. The details of the GSM authentication procedure are hidden in this sequence diagram.

4 Conclusion and future works

This paper proposes the novel design of a unified authentication system for services accessed through a mobile handset. Specifically, the proposed solution makes use of the already ubiquitous authentication mechanism provided by existing GSM/UMTS networks.

By combining the GSM SIM authentication mechanisms with the EAP-SIM framework, we achieve mutual authentication between the parties. Since the SIM is a tamper resistant Smart Card, and the user has to present a valid PIN to activate the SIM, we have also achieved strong two-factor authentication, which fulfils the highest security level defined by NIST [12]. So we got a secure system, which is also easy to use, since the user only needs one single device, the mobile handset, to access the secure services. One of the greatest benefits is that the proposed system is generic, which means it can be used by both mobile browsers and stand-alone applications on the mobile handset.

Parts of the proposed system have already been implemented, and most of the external components needed already exist in the GSM network today. One challenge might be to realize the Supplicant as a local proxy on the mobile handset and to be able to run it on a broad range of mobile handsets. Another possible challenge is the communication with the SIM through the SATSA-APDU package. According to the specifications, there should be no major problems to extract the user credentials from the SIM, as long as the Supplicant MIDlet is signed with a certificate issued by the telecom operator. However, due to vague specifications, some challenges might arise which have not been predicted yet.

Integrating the solution with an Identity Management System (e.g. Liberty Alliance) [5] could be the next step for this project. To be able to offer security services in cooperation with many different types of Service Providers with proprietary technologies that do not interoperate easily, a standardized methodology for exchanging authentication data between security domains is required.

By adopting the SAML framework [13] or similar, it will be possible to offer the proposed system as a part of a single sign-on system (SSO) in the future.

Another possible enhancement is to utilize the GSM cipher key (Kc) for encryption purposes. The information exchange between the client application and the Service Provider must be protected by some means, and by utilizing Kc in a lightweight crypto package like Bouncy Castle [14], we can achieve end-to-end encryption without the bothersome key exchange, since Kc can be derived from the SIM.

References

1. WAP Forum, WAP-260-WAP Identity Module (WIM), 2001 http://www.wapforum.org/tech/documents/WAP-260-WIM-20010712-a.pdf
2. WTLS Specification WAP-199, http://www.wapforum.org/tech/documents/WAP-199-WTLS-20000218-a.pdf
3. ISO/IEC 7816, Smart Card Standard, http://www.iso.org
4. Do, T.V. et. al. (2006). "Offering SIM Strong Authentication to Internet Services", whitepaper, 3GSM 2006, http://www.simstrong.org/resources.php
5. The Liberty Alliance, http://www.projectliberty.org
6. Sun Microsystems, SATSA Developers Guide 1.0, 2004 http://sw.nokia.com/id/2e279a27-26ef-4435-8492-9ebae977aa9c/MIDP_SATSA_APDU_API_Developers_Guide_v1_0_en.pdf
7. B. Aboba ET.AL., IETF, RFC3748 - Extensible Authentication Protocol (EAP), 2004 http://www.faqs.org/rfcs/rfc3748.html
8. H. Haverinen ET.AL.,IETF, RFC4186 – Extensible Authentication Protocol Method for GSM SIM, 2006 http://www.faqs.org/rfcs/rfc4186.html
9. B. Aboba ET.AL., IETF, RFC3579 - Remote Authentication Dial In User Service (RADIUS) support for EAP, 2003. http://www.faqs.org/rfcs/rfc3579.html
10. Ortiz Enrique, The MIDP 2.0 Push Registry, January 2003 http://developers.sun.com/techtopics/mobility/midp/articles/pushreg/
11. Kay Glahn ET.AL, JSR 248: Mobile Service Architecture, 2006 http://jcp.org/en/jsr/detail?id=248
12. Burr, E William ET.AL, NIST (2006) Electronic Authentication Guideline, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
13. OASIS Security Services, Security Assertion Markup Language (SAML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
14. Bouncy Castle Crypto APIs for Java, http://www.bouncycastle.org

Index